



Tommi Eirola

ÄLYKKÄÄT KODINKONEET

ÄLYKKÄÄT KODINKONEET

Tommi Eirola
Opinnäytetyö
Syksy 2013
Tietotekniikan koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, langattomat laitteet

Tekijä(t): Tommi Eirola
Opinnäytetyön nimi: Älykkäät kodinkoneet
Työn ohjaaja(t): Veijo Korhonen (OAMK)
Työn valmistumislukukausi ja -vuosi: Syksy 2013
Sivumäärä: 36 + 50 liitettä

Opinnäytetyö toteutettiin Oulun seudun ammattikorkeakoulun toimesta. Työn aiheena oli tutkia Smart Grid ja Smart Home -teknologioita ja hyödyntää niitä testiympäristössä. Smart Grid tarkoittaa älykästä sähköverkkoa, jossa tiedonsiirto on kaksisuuntaista eli se kulkee sekä sähkötoimittajalta asiakkaalle että päinvastoin. Smart Home tarkoittaa rakennusta, jossa laitteet pystyvät kommunikoidaan keskenään ja tiedonsiirto kulkee esimerkiksi PLC:n avulla pistorasioita pitkin huoneesta toiseen. Työssä jatkettiin Marko Vesalan tekemää opinnäytetyötä, jossa hän oli tutkinut ja rakentanut PLC-yhteydet ja tehnyt testiympäristön tälle.

Työn tavoitteena oli tutkia Arduinon toimintaa ja sen hyödyntämistä Smart Home -teknologiassa. Työssä oli tarkoitus rakentaa jääkaapin toimintoja, kuten energian kulutusta seuraava laite, jolla pystyisi myös ohjaamaan sen toimintoja Internetin välityksellä. Työssä käytettiin Marko Vesalan opinnäytetyötä varten tilattuja ZyXEL PLA4211, ZyXEL PLA4201 PLC-testaukseen sekä testiympäristöön Arduino Mega 2560 ja Arduino Ethernet Shield -piirilevyjä.

Työssä ei pystytty tekemään pääaihetta eli jääkaapin kulutuksen seuraamista ja sen päälle ja pois kytkemistä, koska siihen tarvittavaa laitteistoa ei saatu käyttöön vielä. Työssä kuitenkin saavutettiin nämä testiympäristön avulla eli Internet-sivun kautta toimiva päälle/pois-kytkin, joka sytytti ja sammutti LED-valon, sekä se, että Internet-sivulta pystyi seuraamaan jääkaapin oven aukaisuja ja lämpötilaa. Ongelmaksi koituivat Arduinon rajalliset käyttöominaisuudet. Esimerkiksi hälytyksen alkamisajankohtaa ei pystytty tekemään sillä tavalla, että asiakas voisi itse määrätä, koska Arduino-ohjelmointiympäristö ei tue tarvittavaa Javascriptiä.

Jatkossa kannattaisi vaihtaa Arduino useampaa ohjelmointikieltä tukevaan alustaan. Tämä edesauttaisi jatkokehitystarpeita, joissa ohjelmaa voitaisiin jatkaa muihinkin laitteisiin ja Internet-sivujen parannus olisi helpompaa.

Asiasanat: PLC, Arduino, Smart Grid, Smart Home

ABSTRACT

Oulu University of Applied Sciences
Information technology and Telecommunications, Wireless devices

Author(s): Tommi Eirola
Title of thesis: Smart Home Appliances
Supervisor(s): Veijo Korhonen (OUAS)
Term and year when the thesis was submitted: Autumn 2013
Pages: 36 + 50 appendices

This bachelor's thesis was assigned by Oulu University of Applied Sciences. The subject of the thesis was to research Smart Grid and Smart Home concepts and utilize them in the test environment. The Smart Grid means grid where the data transfer is two-way, which means that the data transfer goes from the supplier of electricity to the customer and the other way round. The Smart Home connect all the devices and the appliances in your home so they can talk to each other for example with PLC but there is also other options for that. The thesis was continuing Marko Vesala's thesis where he had researched and built PLC connections and made a test environment for this.

The aim of this thesis was to examine Arduino's properties and utilize them in Smart Home technology. Purpose of thesis was to build a device to follow the parameters of a fridge which might also be able to control its functions over the Internet. In the thesis was used ZyXEL PLA4211, PLA 4201 for testing of PLC connections and Arduino Mega 2560 with Ethernet Shield for the test environment.

Main subject of the thesis was not possible to reach, which would have been to follow energy consumption of the fridge and switching the power on and off because the necessary equipment was not made available yet. However, in the thesis these accomplishments were achieved by using the test environment which turns a LED off/on and follows the opening of the door and the temperature of the fridge via a Web page. Another big problem was a limited performance of Arduino. For example, the timeout when an alarm begin, could not be done such a way that the customer could set it by himself/herself because Arduino programming environment does not support required Javascript.

In the future Arduino should be replaced with another device which could be used more freely with different programming languages. This would provide advantages for further development in which the program could be extended to other home applications and the development of web page would be easier.

Keywords: PLC, Arduino, Smart Grid, Smart Home

ALKULAUSE

Tämä opinnäytetyö tehtiin syksyllä 2013 Oulun seudun ammattikorkeakoulun toimeksiannosta. Työn ohjaajana toimi lehtori Veijo Korhonen ja työn tilaajana Henry Hinkula.

Haluaisin kiittää Veijo Korhosta opinnäytetyön idean kehittäjänä ja mahdollisuudesta päästä tekemään sitä. Työn ohjaamisesta haluaisin kiittää Henry Hinkulaa. Hänestä oli paljon apua työn loppuun viemisessä. Myös Tommi Sallinen oli apuna työtä tehdessä. Kiitokset kuuluvat myös Marko Vesalalle työn aloituksesta sekä myös uusien ideoiden antamisesta.

Oulussa 16.12.2013

Tommi Eirola

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
ALKULAUSE	5
SISÄLLYS	6
SANASTO	7
1 JOHDANTO	8
2 SMART HOME ELI ÄLYKÄS KOTI	9
2.1 Energian säästö	10
2.2 Sähkökatkot	12
3 PLC	16
4 TIETOTURVA	17
5 TOTEUTUS	19
5.1 Arduino	20
5.1.1 Arduino-piirilevy	21
5.1.2 Arduino-ohjelmointityökalu	21
5.2 Työn aloittaminen	23
5.3 Ohjelman pilkkomista ja ovikytkimen asentaminen	27
5.4 Tietokanta	28
5.5 Lopullinen tuote	30
6 TULOKSET JA POHDINTA	32
LÄHTEET	34
LIITTEET	36

SANASTO

Smart Home	Älykäs koti, jota voidaan ohjata internetin välityksellä.
Smart Grid	Älykäs sähköverkko, jossa tiedonsiirto tapahtuu sekä verkkoyhtiöstä asiakkaan suuntaan että päinvastoin.
PLC	Power Line Communication, Sähköjohtoja pitkin tapahtuva tiedonsiirto.
WLAN	Wireless Local Area Network, langaton lähiverkko
PuTTY	Telnet ja SSH-yhteyksien pääte-emulaattori.

1 JOHDANTO

Työni pohjautuu PLC-tekniikkaa hyödyntävään datansiirtoon. Tulevaisuudessa sitä voidaan käyttää älykkäissä rakennuksissa, joissa esimerkiksi sähkönkulutusta seurataan tunneittain. Eli sähkönkulutusta lisätään kun sähkönhintaa on halvimmillaan, kun taas sähkönhintaa on kalliimpaa, yritetään käyttää laitteita minimikulutuksella, jotta laitteen käyttäminen olisi mahdollisimman halpaa. Tällaisia laitteita ovat esimerkiksi jääkaapit, johon myös oma työni kohdistuu.

Työssäni jatkan aiemman opinnäytetyön tekijän, Marko Vesalan, tekemää työtä, jossa hän oli rakentanut PLC-verkon valmiiksi ja tutkinut niiden ominaisuuksia. Lisäksi hän oli tehnyt pohjan omalle ohjelmistolleni ja tehnyt testiversion sähkön- ja lämmönmittauksille.

Työssä käytetään Arduino-ohjelmointityökalua, joka on synkronoitu toimimaan Arduino-piirilevyn kanssa. Lisäksi työssä käytetään WAMPServer-tietokanta ohjelmaa, johon tallennetaan mittauksien tuloksia, joista tulostetaan diagrammeja käyttäjälle.

2 SMART HOME ELI ÄLYKÄS KOTI

Smart Home eli älykäs koti tarkoittaa nykyaikaista asuin- tai toimistorakennusta, missä laitteet voivat kommunikoida keskenään. Kotiverkon toteutusvaihtoehtoja on useita esimerkiksi Ethernet-kaapelilla tai WLAN:lla. Tässä työssä tiedonsiirto on siirretty normaaleista Ethernet-kaapeleista ja WiFi-yhteyksistä sähköverkkoon muodostaen siten kotiverkon. Tämä tapahtuu kahden tai useamman PLC-adaptereiden avulla, jotka kommunikoivat keskenään ja siirtävät tietoa sähköverkon kautta. Älykäs koti käyttää tätä ominaisuutta hyväksi säädelläkseen kodinkoneita kotiverkon kautta.

Nykypäivän ”älykäs koti” kehittyy vauhdilla ja sillä on useita ominaisuuksia. Aiemmin kodin automatisointi ja hallinta oli yleisesti perustettu erilleen ja kodin ohjaus verkkoon tapahtui omilla viestintä-protokollilla, -ohjaimilla ja -sovelluksilla. Älypuhelimien ja tablettien tulon jälkeen yhä useammat asiakkaat halusivat kontrolloida heidän älykotiansa mobiililaitteillaan ja seurata sen tilaa, kun he ovat poissa kotoa. Itsestään selvä ratkaisu tähän oli yhdistää kodin laajakaista-yhteys ja kodin ohjausverkko toimimaan saumattomasti sähkölaitteiden kontrollointiin. (1.)

Koska nykyajan kodeista on tulossa yhä enemmän viihdekeskuksia, tarvitaan niihin tehokas kotiverkko. Videoiden streamaus eli suoratoisto TV:ssä tai tietokoneella, pelien pelaamista netissä, HDTV ja useat muut laitteet, jotka käyttävät laajakaistaa sovelluksien pyörittämiseen, kuormittavat valtavasti kotiverkkoa. Vaikka langaton laajakaista tarjoaa hyvän liikkuvuuden ympäri taloa, yleensä se kuitenkin toimii huonosti jos reititin on kaukana laitteesta, johon sitä tarvitaan. Tällöin erityisesti vaativat sovellukset, kuten verkkopelaaminen, tuottavat ongelmia. (1.)

Jotta Smart Home toimisi halutulla tavalla ja se olisi mahdollisimman hyödyllinen sekä säästäisi sähköä, on siihen liitettävä älykäs sähköverkko eli Smart Grid. Smart Grid on sekä toimintavarma että ympäristöystävällinen sähköverkko, koska sillä pystytään paikantamaan mahdolliset sähkökatkokset nopeasti sekä se antaa tietoa kulutuksen määrästä. Tällöin alueille, joissa on

suuri kulutus, voidaan rakentaa hajautettua pientuotantoa. Energiateollisuuden Internet-sivun artikkelissa kuvataan älykästä sähköverkkoa seuraavasti:

Älykkäälle sähköverkolle on olennaista kaksisuuntaisuus. Hajautetun tuotannon myötä energia virtaa nykytilanteesta poiketen molempiin suuntiin. Samoin tiedonsiirto tapahtuu sekä verkkoyhtiöstä asiakkaan suuntaan että päinvastoin. Tiedonsiirto mahdollistaa myös verkonhallinnan verkon käyttötapojen muuttuessa.

Älykäs sähköverkko on edellytys EU:n ympäristötavoitteiden toteutumiselle. Älykäs sähköverkko tukee myös komission sähkömarkkinoiden integraatiolle, kilpailukyvyille, toimitusvarmuudelle asettamia tavoitteita. Älykkään sähköverkon odotetaan lisäävän kansalaisten hyvinvointia ja luovan paljon uusia työpaikkoja. Älykäs sähköverkko auttaa kansalaisia tehostamaan energiankäyttöään. Samalla se mahdollistaa kysyntäjoustot. Pienimuotoisen erityisesti kiinteistökohtaisen mikrotuotannon lisääntyminen johtaa verkon käyttötapojen muuttumiseen. Tämä ei ole mahdollista ilman, että verkon älykkyyttä lisätään. Samalla parantuvat sähköjakelun luotettavuus ja verkon käyttövarmuus.

Älykästä sähköverkkoa on usein kuvattu energian internetiksi. Taustalla on IT-teknologian huima lisääntyminen. Suomessa sähköverkko on jo varsin älykäs. Voidaankin sanoa, että meillä on käytössä jo Smart Grid 1.0. Esimerkkejä verkon älykkyydestä ovat automaattinen vianpaikannus ja -erotus, verkon käytön optimointi ja etäluettavat mittarit. Älykäs energiamittari sekä kaksisuuntainen tiedonsiirto asiakasliittymän ja verkon välillä ovatkin älykkäässä sähköverkossa keskeisessä asemassa. (2.)

2.1 Energian säästö

Työssä oli ajatuksena toteuttaa laite toimimaan tuntihinnoittelun mukaisesti. Tämä on uusi tapa sähköjakelussa, joka mahdollistaa kodinkoneiden käyttämisen silloin, kun sähkönhinta on alhaisempi. Pekka Pantsu kertoo tämän hyvistä puolista, mutta myös sen, mitä on otettava huomioon ennen tällaisen sopimuksen tekemistä. Hän kirjoittaa:

Sähkön tuntihinnoittelu tekee tuloaan kotitalouksiin. Tuntihinnoitteluun perustuva sähkökauppa on käytännössä tullut mahdolliseksi kotitalouksille vasta hiljattain, kun etäluettavat sähkömittarit ovat yleistyneet kodeissa. Samalla sähköyhtiöt lisäävät tuntihinnoittelun yhdeksi vaihtoehdoksi sähkösopimuksiinsa.

Tuntihinnoittelun avulla kuluttaja voi nykyistä paremmin vaikuttaa maksamansa sähkön hintaan. Tässä hinnoittelussa kuluttajan maksama sähkön hinta elää sähköpörssin tahdissa. Hinta voi myös kallistua, nopeastikin. Riskin siitä kantaa kuluttaja. Hintavaihtelut vaikuttavat kuluttajan normaaleihin arjen askareisiin. Hän valitsee esimerkiksi pyykkäämisen ja saunomisen ajankohdan tarkemmin kuin ennen. (3a.)

Sähköntuntihinnoittelu vaikuttaisi myös sähköntoimittajiin myönteisesti.

Fortumin Internet-sivulla on otettu tähän kantaa sähköntoimittajien kannalta:

Kulutuksen ohjaaminen halvimmille tunneille tasapainottaa myös sähköverkon kuormitusta ohjaamalla kulutusta automaattisesti pois huipputunneilta. Halvimmilla tunneilla tuotettu sähkö on myös ympäristön kannalta vähäpäästöisin vaihtoehto, koska kulutushuippujen sähköntarve katetaan useimmiten fossiilisilla polttoaineilla. (3b.)

Alla olevan taulukon (taulukko 1) mukaan asiakas voi suunnitella päivittäisen sähkönkulutuksen. Taulukko päivittyy joka päivä ja se näyttää myös seuraavan vuorokauden ennakkolukemat. Taulukosta huomataan, että esimerkiksi sähkölämmitteinen asunto kannattaisi lämmittää yöllä ja päivällä lämmitys laitettaisiin pois päältä tai minimilämmitykselle.

TAULUKKO 1. Sähkön hinta tunneittain Suomessa (3c.)

Elspot price for area Finland in EUR/MWh								
	12-12-2013	11-12-2013	10-12-2013	09-12-2013	08-12-2013	07-12-2013	06-12-2013	05-12-2013
00 - 01	32,21	33,48	34,91	34,10	35,99	33,57	32,42	31,06
01 - 02	31,69	32,09	33,93	33,95	34,93	33,24	31,92	30,28
02 - 03	31,51	32,06	33,17	33,91	34,34	33,08	31,27	29,67
03 - 04	31,55	32,18	33,19	34,00	34,20	33,14	31,43	30,32
04 - 05	31,95	32,55	34,16	34,54	34,64	33,42	32,20	31,52
05 - 06	36,94	35,47	52,59	45,06	35,74	33,66	33,12	35,02
06 - 07	42,91	46,25	67,57	73,60	36,49	33,96	34,09	43,08
07 - 08	45,55	48,77	74,11	109,93	37,40	33,91	36,32	45,85
08 - 09	42,07	43,95	67,15	78,43	37,91	34,92	36,86	45,94
09 - 10	35,86	37,80	66,55	47,72	39,04	36,32	36,25	37,62
10 - 11	36,65	37,32	50,98	43,82	39,76	35,76	35,50	35,89
11 - 12	35,55	36,95	46,65	44,58	40,84	36,80	35,34	35,40
12 - 13	34,79	36,68	47,84	45,10	39,33	36,61	34,83	35,83
13 - 14	37,45	38,02	60,02	60,63	39,94	36,81	34,75	39,29
14 - 15	40,00	38,49	59,91	79,18	45,06	39,96	35,12	44,28
15 - 16	43,82	42,94	60,32	150,01	52,76	50,01	39,94	45,78
16 - 17	47,06	43,36	60,08	71,43	53,08	50,07	42,06	49,04
17 - 18	47,04	43,69	60,41	94,30	53,04	50,05	40,91	45,71
18 - 19	37,34	37,61	52,51	77,49	53,02	43,94	36,34	37,63
19 - 20	34,33	36,59	38,91	59,94	45,10	40,69	34,53	36,99
20 - 21	33,94	36,16	36,73	45,42	41,41	38,86	33,78	35,94
21 - 22	33,53	35,85	36,00	42,43	39,95	39,20	33,77	33,99
22 - 23	32,67	34,97	35,02	37,76	36,90	37,59	33,39	33,43
23 - 00	31,64	33,96	34,38	35,55	35,40	35,95	33,09	32,89
Min	31,51	32,06	33,17	33,91	34,20	33,08	31,27	29,67
Max	47,06	48,77	74,11	150,01	53,08	50,07	42,06	49,04
Average	37,00	37,80	49,05	58,87	40,68	37,98	34,97	37,60
Peak	39,33	39,45	55,94	71,05	44,91	41,00	36,87	40,78
Off-peak 1	35,54	36,61	45,45	49,89	35,47	33,50	32,85	34,60
Off-peak 2	32,95	35,24	35,53	40,29	38,42	37,90	33,51	34,06

2.2 Sähkökatkot

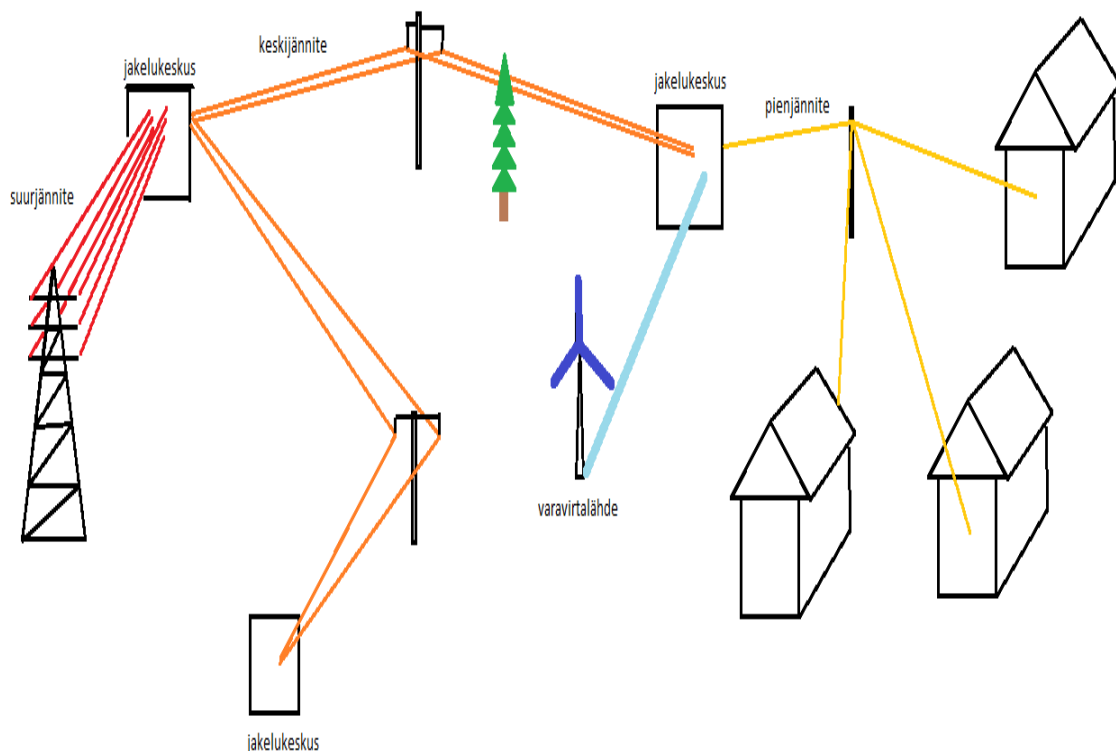
Sähkökatkoja tapahtuu muutamia kertoja vuodessa johtuen yleensä luonnonilmiöistä. Nämä vaikuttavat sähköntoimittajien talouteen, koska niiden korjaamiseen ja korvaamiseen kuluu rahaa. Energiatalouden Internet-sivuilla kerrotaan mahdollisista sähkökatkon aiheuttajista sekä niiden vaikutuksista:

Sähkön käyttäjän kannalta sähkönjakelun toimitusvarmuus on avainasemassa. Sähkökatkon myötä valot sammuvat, sähkölämmitys lakkaa toimimasta ja sähköllä toimivia kodinkoneita ei voi käyttää. Kaupungeissa pysähtyvät muun muassa metroliikenne, hissit, kauppojen kassat ja sähköovet. Myös vesi- ja jätehuolto saattavat keskeytyä jo muutaman tunnin sähkökatkoista.

Sähkökatkoja aiheuttavat luonnontapahtumat ja tekniset viat. Sään sähköverkolle aiheuttamia riskejä ovat johdoille kaatuvat puut, salamet, lumi- ja jääkuormat, tulvat ja kova pakkanen. Myös katkon pituuteen ja laajuuteen luonto vaikuttaa oleellisesti. Teknisistä vioista aiheutuvat keskeytykset eivät yleensä ole pitkäaikaisia eivätkä laajoja, sillä vika-alue voidaan eristää muusta verkosta. Kovien myrskyjen aikaan puolestaan vikoja esiintyy paljon yhtä aikaa ja korjaustyöt ovat muutenkin hankalia. Kustannukset ja katkojen määrät myrskyistä ovatkin nousseet viime aikoina korkeiksi. (4.)

Energiatalouden Internet-sivulla on esitetty tähän ratkaisuja. Ne vähentäisivät huomattavasti sähkökatkoista aiheutuvia haittoja:

Valtaosa keskeytyksiä aiheuttavista vioista tapahtuu ilmassa kulkevalla avojohdoilla. Voimajohtojen kaivaminen maan alle eli kaapelointi olisi paras keino parantaa sähkön toimitusvarmuutta. Maahan kaivaminen on kuitenkin ilmajohdon rakentamista kalliimpaa. Toinen parannuskeino on varasyöttöjen ja rengasyhteyksien rakentaminen. Sähkön jakeluverkkoja käytetään yleensä säteittäisinä eli siten, että kuhunkin kohteeseen tulee vain yksi johto. Tiettyihin kohteisiin voidaan kuitenkin rakentaa varasyöttöyhteyksiä. Kun jokin johto vikaantuu, kohteeseen tuleva varasyöttö otetaan käyttöön. (4.)



KUVA 1. Varasyöttö

Kuva 1 esittää varasyöttöyhteyttä, kun puu on kaatunut pääsyöttöyhteyden päälle ja aiheuttanut sähkökatkoksen. Tässä tapauksessa varasyöttönä toimii tuulivoimala. Normaalisti varasyöttö tapahtuu muiden jakelukeskuksien ja keskijännitekaapeleiden kautta, mutta tässä kuvassa on kehitelty toisenlainen ratkaisu. Varasyöttöä voisi myös käyttää silloin, kun päävirransyöttö ylikuormittuu eli silloin kun asiakkaista johtuva kulutus kasvaa. Normaalisti jouduttaisiin käyttämään lisäsyöttöä fossiilisten apukeinojen avulla, mutta jos jokaiseen jakelukeskukseen olisi liitetty tällainen varasyöttöasema, ei niitä tarvitsisi käyttää. Tällaisia varasyöttöasemia ovat tuuli-, vesi- ja biovoimalat sekä aurinkopaneeleista tuleva sähkö. Tällaista varasyöttöä kutsutaan hajautetuksi pientuotannoksi. Energiatieteellisyys kertoo hajautetusta pientuotannosta seuraavaa:

Pienimuotoista hajautettua tuotantoa tullaan liittämään enenevässä määrin sähköjakeluverkkoon. EU:n uusiutuvan energian lisäys- ja ilmastotavoitteet, pienten tuotantolaitosten hintojen halpeneminen, kuluttajien halu pienentää sähkölaskuaan sekä kulutuspäätöksiä enenevästi ohjaava ympäristötietoisuus ovat lisänneet kiinnostusta pienimuotoista tuotantoa kohtaan. Olosuhteiltaan sopivalla paikalla sijaitset omakotitalot, maatalot tai pienyritykset voivatkin harkita tuottavansa sähköä lähinnä omiin tarpeisiinsa esim. pientuulivoimalan tai aurinkopaneelien avulla. (5.)

Hajautettu pientuotanto vähentäisi huomattavasti sähkökatkoista aiheutuvia haittoja. Mutta tämä vaatisi kuorman hallintaa esimerkiksi älykkäiden kodinkoneiden avulla, koska syöttöteho on pieni, johon Energiatalous kommentoi:

Verkonhaltijan tehtävänä on tarjota tuotannolle luotettava verkko ja taata sähköjakelun toiminta ja turvallisuus kaikille verkon käyttäjille myös tuotantolaitoksen liittämisen jälkeen. (5.)

Ennen kuin saataisiin kattava hajautettu pientuotantoverkosto, vaadittaisiin siihen isoja investointeja sekä sopimuksia paikallisten verkkoyhtiöiden kanssa. Energiatalous kertoo sopimuksista sekä niihin liittyvistä standardeista näin:

Tuotantolaitoksen sähköverkkoon liittämistä suunnittelevan tulee tarkistaa paikalliselta verkkoyhtiöltä tuotantolaitoksen verkon liittämistä koskevat velvoitteet ja ohjeet. Tuotantolaitosten sähköverkkoliitännän ja sähköisten ominaisuuksien tulee täyttää

sähköturvallisuusstandardien sekä sähkömagneettisten yhteensopivuusstandardien vaatimukset. Standardien mukaisilla laitteilla varmistetaan, ettei sähköverkossa työskenteleville asentajille aiheudu hengenvaaraa ja että verkon jännitteen laatu pysyy riittävän hyvänä tuottajalle itselleen sekä naapureille. Tuotantolaitoksen asennustyöt saa tehdä vain sähköalan ammattilainen. (5.)

Hajautettua pientuotantoa voisi hyödyntää myös sillä tavoin, että kun kuluttaja tietää ajankohdan, milloin sähköä kuluisi enemmän esimerkiksi saunan lämmitykseen, voisi hän kilpailuttaa sähköntoimittajia etukäteen. Tällöin tällaiseen sähkön kuormitukseen osattaisiin varautua ja ohjata sähkö tulemaan varasyöttölähteestä.

3 PLC

PLC eli Power Line Communication tarkoittaa rakennuksen sisäisessä sähköverkossa tapahtuvaa tietoliikennettä. Tätä varten tarvitaan PLC-adapterit, jotka kommunikoivat keskenään. Internet-yhteys näihin adaptereihin saadaan modeemin kautta Ethernet-kaapelin avulla ja itse adapteri laitetaan läheisimpään pistorasiaan (kuva 2). Jotta Internet-yhteys saadaan toiseen huoneeseen, tarvitaan toinen adapteri, josta niin ikään Internet-yhteys vietään laitteeseen Ethernet-kaapelilla.

PLC-adapterit mahdollistavat siis Internet-yhteyden siirron toisiin huoneisiin ilman ylimääräisiä johtoja. Tarvitaan vain samassa vaiheessa olevat sähkölinjat, jotta yhteys toimisi.



KUVA 2. PLC-adapteri on liitetty sähköverkkoon

4 TIETOTURVA

Nykyään tietoturva-asiat ovat nousseet esille joka puolella maailmaa, erityisesti isoissa ja tärkeissä yrityksissä, mutta myös pienissä yrityksissä ja kodeissa. Viime aikoina tietoturvaan liittyvät murtautumiset ovat yleistyneet ja arkaluontoisia asioita on tullut esille. Myös suuria summia rahoja on viety, koska tietoturva-asiat eivät ole olleet kunnossa. Myös murtautujat ovat kehittyneet, joten tähän asiaan on paneuduttava jatkossa yhä enemmän ja enemmän. On kehiteltävä parempia keinoja salasanojen ja käyttäjätunnuksien turvaamiseen. Suomen Internetopas määrittelee tietoturvan seuraavasti:

Tietoturvalla pyritään suojaamaan yritykselle tärkeät tiedot ulkopuolisilta. Kysymys on siis toimenpiteistä, jotka takaavat yhtiön tietojen koskemattomuuden. Tietoturvalla on asetettu tiettyjä tavoitteita: niitä ovat yksilön tai organisaation annettujen tietojen luottamuksellisuus, eheys, kiistämättömyys, pääsynvalvonta, saatavuus ja tarkastettavuus. Jotta tiedot ovat luottamuksellisia, niiden on oltava vain niihin oikeutettujen käytössä. Tiedoilla ja dokumenteilla on turvaluokitus, joka määrittelee sen, kenellä on oikeus tietojen käyttöön sekä säilytykseen ja tuhoamiseen. (6.)

Tietoturvaongelman voisi ratkaista käyttäjätunnuksilla, jotka tallennettaisiin tietokantaan. Jokaisella käyttäjällä olisi siis oma käyttäjätunnus ja salasana. Tämä vaatisi myös perehtymistä salasanan salaamiseen, ettei kukaan pääse käsiksi niihin. Myös tietokanta pitäisi pystyä salaamaan ja piilottamaan niin hyvin, ettei kukaan pääse murtautumaan sinne. Ohjelmasta tekee turvallisen se, miten hyvin tietoturvaan paneudutaan ja kuinka monimutkainen salausjärjestelmä ohjelmaan kehitellään. Seuraavassa on esitelty, mitä Suomen Internetoppaan mukaan tulee ottaa huomioon, kun tietoturvaa käsitellään:

Eheys, alkuperäisen termin mukaan integrity eli oikeellisuus, tarkoittaa tiedon muuttumattomuutta tiedon luomisen, käsittelyn ja siirron aikana. Kiistämättömyydellä valvotaan tiedon siirtoon tai käsittelyyn osallistuneiden käyttäjien tunnistamista. Tällä varmistetaan, ettei kukaan voi käsitellä tietoja huomaamatta. Pääsynvalvonta tarkoittaa, että käyttäjien tietoon käsiksi pääsyä valvotaan ja rajoitetaan. Saatavuudella tarkoitetaan tiedon helppoa ja viiveetöntä käyttöä niille, joilla on siihen oikeus. Tarkastettavuus tarkoittaa, että tietojenkäsittelyn tuloksena saatu tieto on kyettävä tarkastamaan ja sen oikeellisuus kyettävä osoittamaan.

Verkkoympäristössä on mahdollista muun muassa muunnella WWW-sivustojen tietoja luvatta, ohjata Internet-hakuja väärin osoitteisiin, lähettää sähköpostia tekaistuilla nimillä ja kaivaa esiin toisten käyttäjien käyttäjätunnuksia, salasanoja sekä luottokorttien numeroita. Huonosti toteutetusta tietoturvasta mahdollisia seurauksia ovat muun muassa virusten tai ulkopuolisten henkilöiden tunkeutumiset tietojärjestelmiin ja niiden sisältämien tietojen väärinkäytökset tai ilkeät tuhoamiset. (6.)

Koska laite säätää asiakkaan asunnon sähkölaitteita, on erittäin tärkeätä, että lopullisen tuotteen tietoturva on kunnossa, jos ohjelma laitettaisiin Internetiin. Jos tätä ei hoideta kuntoon, kuka tahansa voi sammuttaa esimerkiksi jääkaapin milloin tahansa. Tästä voi syntyä asiakkaalle suuretkin vahingot, joihin toki vaikuttaa, mistä sähkölaitteesta on kyse. Esimerkiksi jos säätö olisi asennettu kasvihuoneen lämmityslaitteeseen, voisi joku käydä sammuttamassa sen, jolloin kasvit voivat kuolla, kun ulkona on liian kylmää.

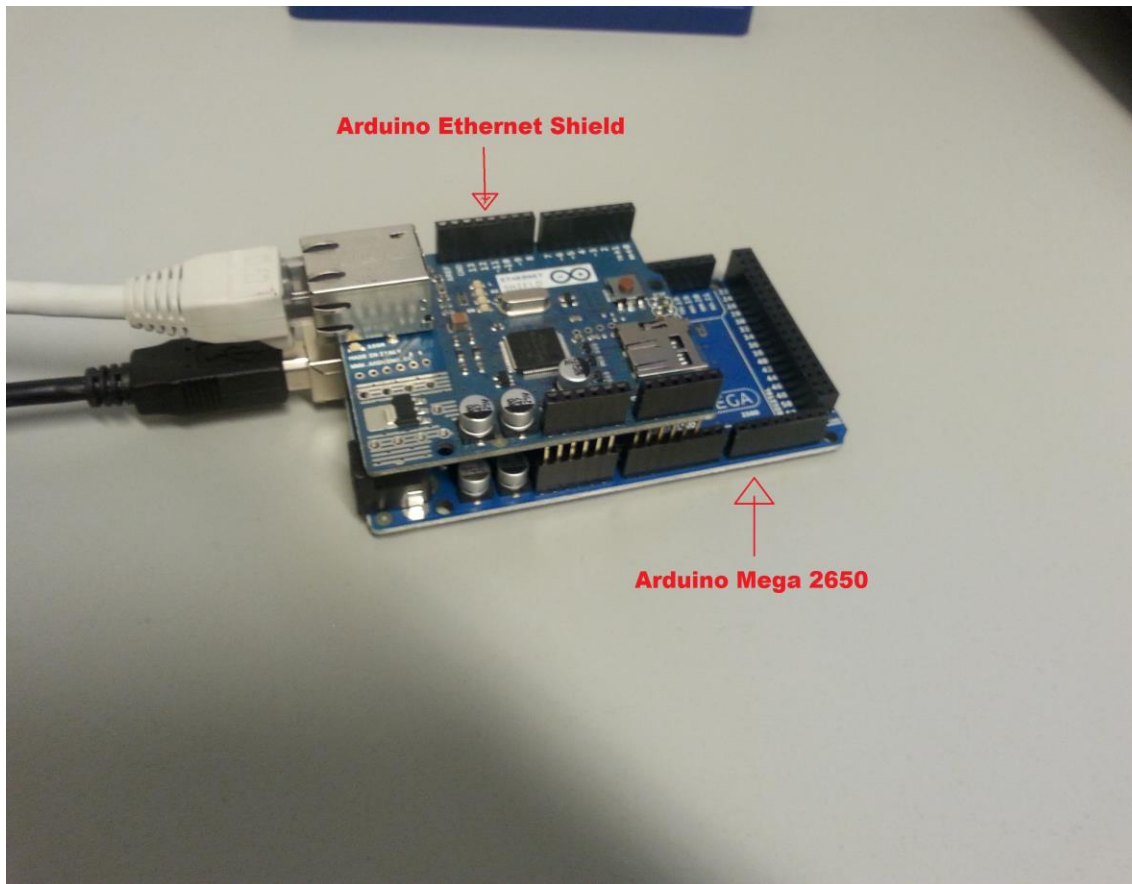
5 TOTEUTUS

Työ tehtiin Oulun seudun ammattikorkeakoulussa opinnäytetyön tekijöille varatussa tilassa. Koulu järjesti minulle työpöydän, tietokoneen ja tarvittavat välineet. Työssäni jatkan Marko Vesalan aloittamaa projektia.

Työ perustuu Smart Home -aihepiirin pohjalle. Tarkoituksena oli rakentaa energiaa säästävää ratkaisua jääkaapille, joka hyödyntäisi sähkön tuntihinnoittelua. Samalla sen toimintaa voisi ohjata etäältä ja se tutkisi jääkaapin tilaa, että se toimii niin kuin sen pitäisi toimia.

Ennen työni aloittamista Marko Vesala oli pystyttänyt PLC-adaptoreiden avulla testiympäristön ja asentanut Arduino Mega 2560 R2 -piirilevyn, johon hän oli liittänyt Arduino Ethernet Shield -piirilevyn (kuva 3), sekä siihen liittänyt lämpöanturin ja potentiometrin eli säätövastuksen. Hän oli myös aloittanut ohjelman tekemisen. Minun työni kohdistuu enimmäkseen ohjelmointityökalulla tehtäviin osioihin, kuten nettisivulta toimiva laitteen ohjaaminen.

Ohjelmointiympäristönä oli Arduino 1.0.5, joka toimii edellä mainittujen piirilevyjen kanssa. Lisäksi komponentteina käytetään punaista LED-indikaattoria, ABI-009-RC piezo-summeria, Reed-ovikytkintä, Dallasin DS1820-lämpötila-anturia ja potentiometriä.



KUVA 3. Arduino-piirilevyt

5.1 Arduino

Arduino-mikrokontrolleri on halpa piirilevy, jolla jopa aloittelijat pystyvät tekemään vaativia ohjelmia ja laitteita. Arduino-piirilevy on yhdistetty Arduino-ohjelmointityökaluun, jossa on oma Arduino-ohjelmointikieli. Piirilevyyn voidaan liittää erilaisia sensoreita, valoja, moottoreita ja muita laitteita, joita ohjataan ohjelmointityökalun avulla. Näistä muodostuu valmiita laitteita, kuten kasvihuoneen lämpötilaa ja kosteutta seuraava ohjelma. (7.)

Erilaisia Arduino-piirilevyjä on kaksikymmentä, joihin voidaan liittää erilaisia piirilevyjä, joissa on lisäominaisuuksia itse pääpiirilevyille. Arduino-piirilevyistä on tehty myös monia kopioita. (7.)

Arduinoja alettiin valmistaa Interaction Design Institute Ivrea -koulussa Italiassa vuonna 2005 opiskelijoiden projekteja varten. Nykyään Arduino on levinnyt ympäri maailmaa. (8.)

5.1.1 Arduino-piirilevy

Arduino-piirilevylle voidaan siis lisätä erilaisia sensoreita, kuten esimerkiksi lämpötilan mittaukseen tarkoitettuja lämpöantureita, kosteusanturi mittaamaan kosteusprosenttia tai LED-valoja ilmaisemaan esimerkiksi hälytystä. Sensorit kiinnitetään piirilevyllä valmiina oleviin pinneihin tai siihen voidaan lisätä erillinen koekytkentäalusta. Koekytkentäalusta on alusta, johon voi liittää erilaisia komponentteja ja kokeilla niiden toimintaa. Tämä on hyvä apuväline projektien suunnittelu- ja valmistusvaiheissa, koska suunnitelmat voivat muuttua ja lopulliseen tuotteeseen tulevaa piirilevyä ei tarvitse tällöin valmistaa vielä tässä vaiheessa.

Itse pääpiirilevyyn voidaan tarvittaessa lisätä lisäosina toimivia piirilevyjä, kuten langattoman Internet-yhteyden mahdollistava WiFi-piirilevy eli se voi toimia itsenäisesti jossain huoneessa ja lähettää sieltä tietoja langattomasti toisessa huoneessa toimivaan tietokoneeseen. Silloin ei tarvita verkkokaapelia, kuten työssämme käytettävässä Ethernet-piirilevyssä.

5.1.2 Arduino-ohjelmointityökalu

Ohjelmointityökalu perustuu C++-ohjelmointikieleen, mutta siinä on pieniä muutoksia. Se on suunniteltu toimimaan Arduino-piirilevyn kautta, eli piirilevy asetetaan johonkin tietokoneen USB-sarjaporttiin, joka valitaan myös ohjelmasta samaksi, jolloin ohjelmointikoodi ja piirilevy toimivat yhtenäisesti. Ohjelmointityökalua ei siis voi käyttää ilman piirilevyä.

Ohjelmointityökalu on hieman yksinkertaisempi kuin muut vastaavat C-kieliset työkalut, koska siihen ei suoraan voi liittää eri ohjelmointikielillä toimivia luokkia. Koodin sisälle voi kuitenkin lisätä toista ohjelmointikieltä, mutta tällöin koodia on hieman vaikeampaa ymmärtää ja joitakin toimintoja ei voi suorittaa normaaliin tapaan. Tällaisia ohjelmointikieliä ovat muun muassa Internet-sivujen rakentamiseen tarkoitettu HTML-kieli ja niiden koristeluun tarkoitettu CSS-kieli.

Arduino-ohjelmointityökalussa on Serial Monitor -työkalu, jolla voi ohjata Arduino-piirilevyä. Tällä työkalulla nähdään myös, toimiiko ohjelma oikein. Jos ohjelmoinnissa on tullut virhe, ohjelma ilmoittaa, minkälainen virhe on

tapahtunut. Työkalun avulla voidaan ohjata ohjelma toimimaan jossain muualla, kuten nettisivulla.

Jotta Arduino-ohjelma toimisi, on siinä oltava pääohjelma, jonka pääte on `ino`, esimerkiksi `main.ino`. Lisäksi pääohjelmassa on oltava sekä `setup()`- että `loop()`-toiminnot. Muuten ohjelma ei osaa suorittaa toimintoja.

`Setup()`-toimintoa kutsutaan, kun ohjelma aloitetaan. Siinä alustetaan kaikki mahdolliset muuttujat ja ulostuloissa olevat sensorit, joita työssä käytetään. Myös Serial Monitor -työkalu alustetaan tässä kohdassa. `Setup()`-toiminto suoritetaan ainoastaan, kun ohjelma aloitetaan tai kun Arduino-piirilevytä painetaan reset-nappia. (9.)

`Setup()`-toiminnon jälkeen suoritetaan `loop()`-toiminto. Siinä suoritetaan kaikki toiminnot, joita työssä käytetään. Toimintoja voi olla useita ja eri paikoissa, mutta niitä täytyy kutsua `loop()`-toiminnon sisällä, jotta ne toimisivat. `Loop()`-toiminnolla kontrolloidaan piirilevyä. (10.)

`Setup()`- ja `loop()`-toiminnot ovat esiteltyinä kuvassa 4.

```

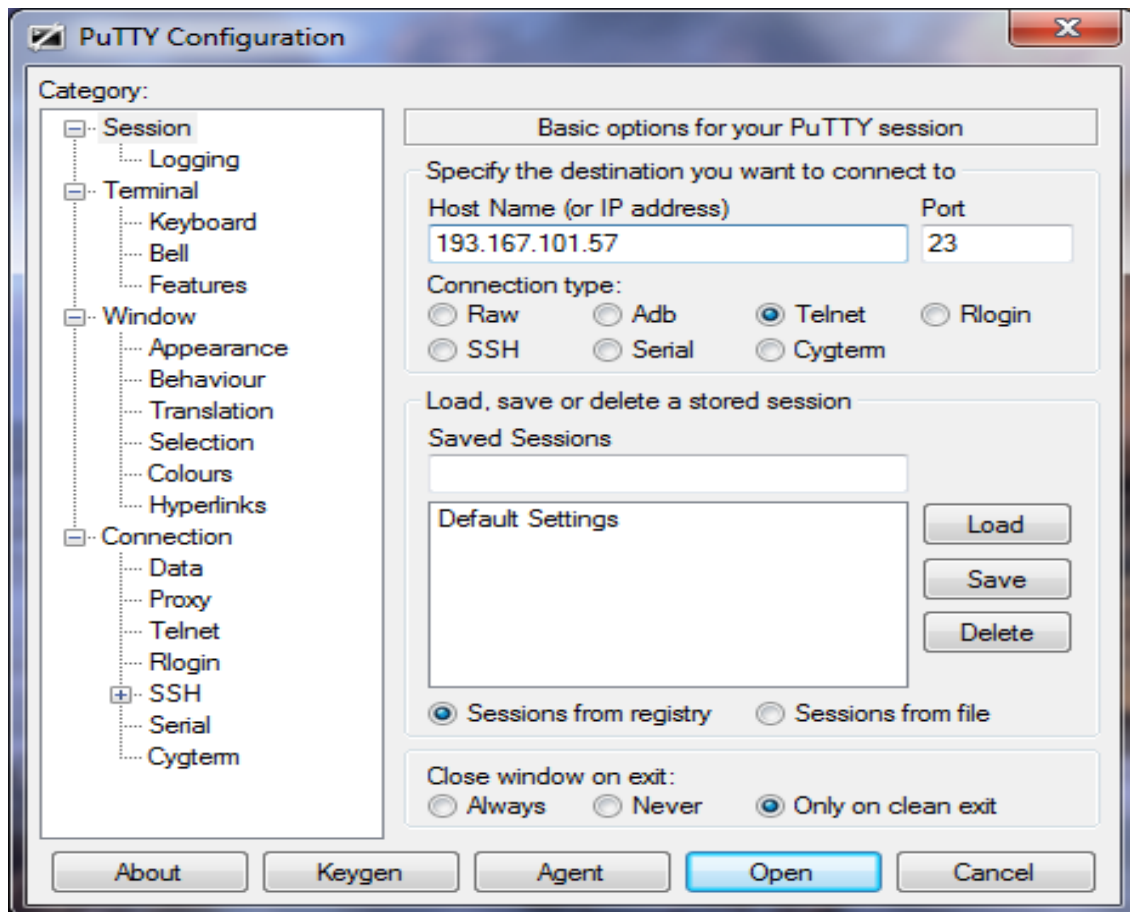
1  const int buttonPin = 3;
2
3  // setup initializes serial and the button pin
4  void setup()
5  {
6      Serial.begin(9600);
7      pinMode(buttonPin, INPUT);
8  }
9
10 // loop checks the button pin each time,
11 // and will send serial if it is pressed
12 void loop()
13 {
14     if (digitalRead(buttonPin) == HIGH)
15         Serial.write('H');
16     else
17         Serial.write('L');
18
19     delay(1000);
20 }

```

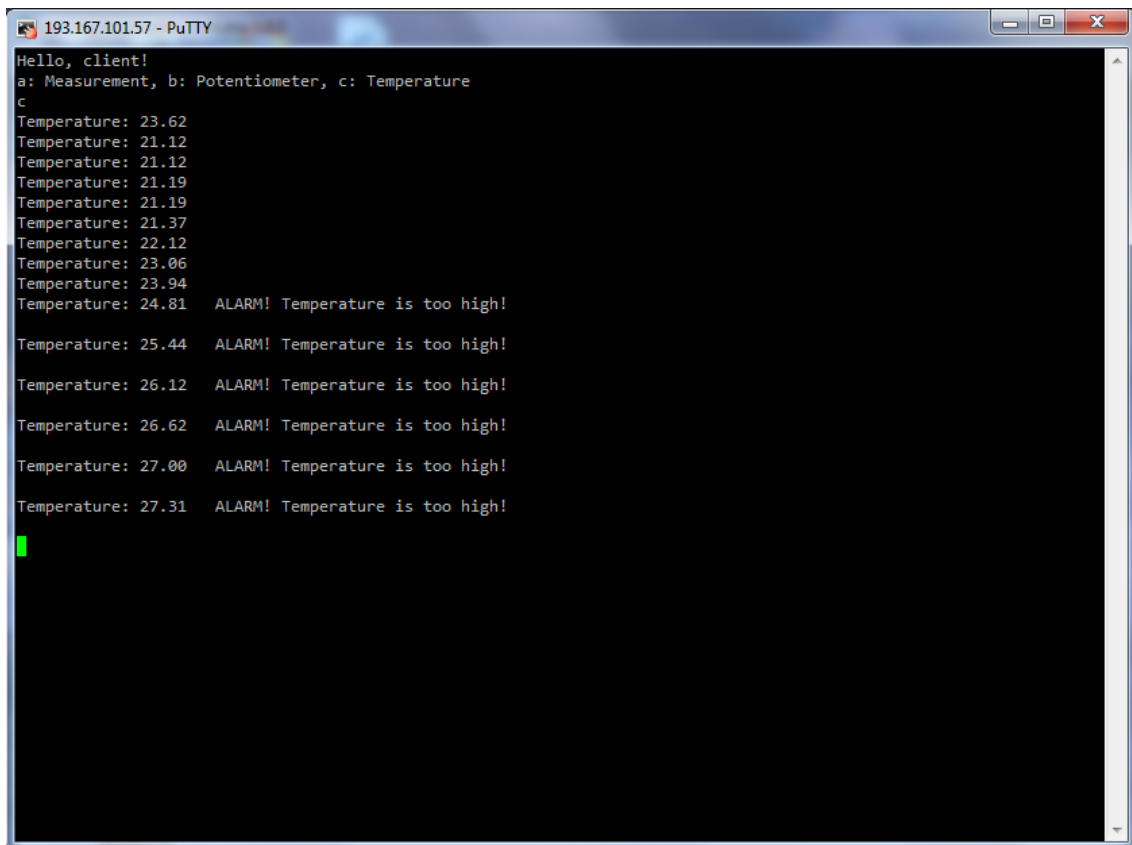
KUVA 4. Esimerkki setup()- ja loop()-toiminnoista

5.2 Työn aloittaminen

Ennen kuin aloin tehdä omaa opinnäytetyötäni, autoin Marko Vesalaa PuTTY-yhteyden valmiiksi saamisessa. PuTTY-yhteys kuvaa tässä tilanteessa kaksisuuntaista tiedonvälitystä. Työssämme Arduino pystyy kommunikoimaan erilaisten laitteiden kanssa samaan aikaan. Esimerkiksi työssämme Arduino voi välittää tietoja sekä PuTTY-yhteydelle että Internet-sivulle. Kuvassa 5 on esimerkki PuTTY-yhteyden luomiseen. Käyttäjän pitää syöttää Arduinon Serial Monitorista saatu IP-osoite sille tarkoitettuun kenttään sekä valita Telnet-yhteys, jolloin portin arvoksi tulee 23. PuTTYllä voidaan lukea potentiometrin, lämpötilan (kuva 6) ja mittauksien arvoja, näitä ovat sähköjännite, -virta ja -teho.



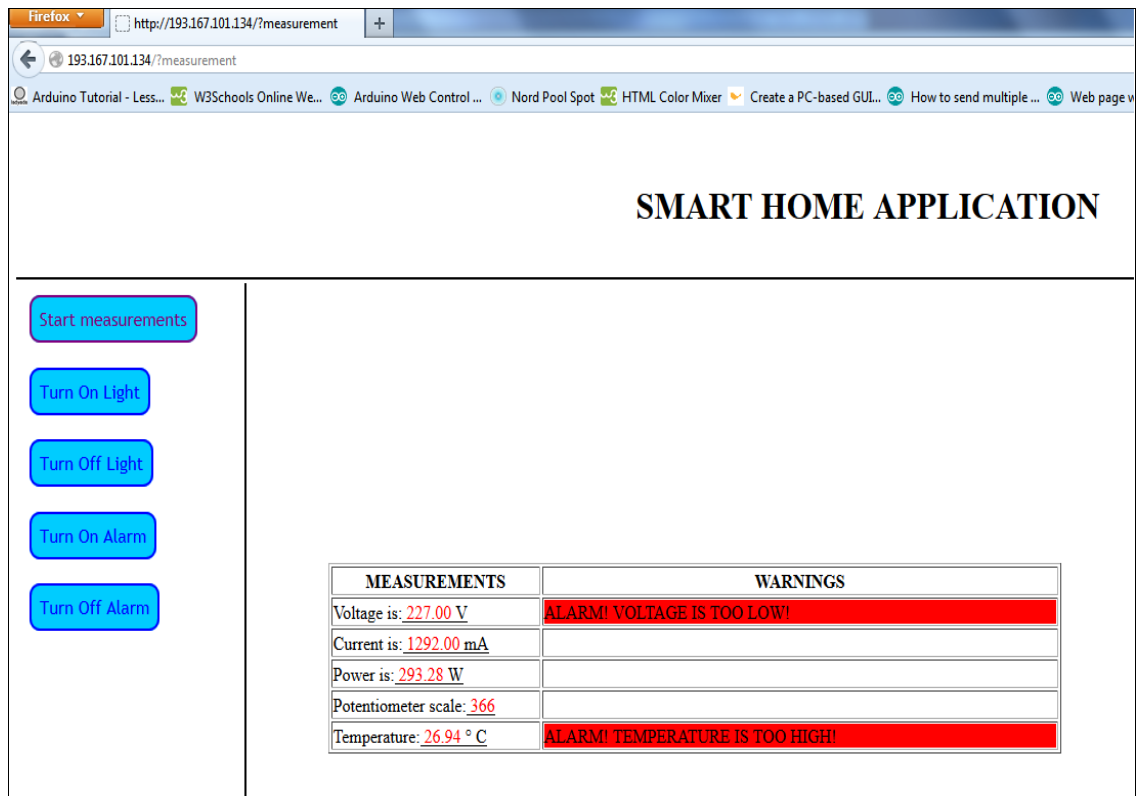
KUVA 5. PuTTY-yhteyden luominen



```
193.167.101.57 - PuTTY
Hello, client!
a: Measurement, b: Potentiometer, c: Temperature
c
Temperature: 23.62
Temperature: 21.12
Temperature: 21.12
Temperature: 21.19
Temperature: 21.19
Temperature: 21.37
Temperature: 22.12
Temperature: 23.06
Temperature: 23.94
Temperature: 24.81  ALARM! Temperature is too high!
Temperature: 25.44  ALARM! Temperature is too high!
Temperature: 26.12  ALARM! Temperature is too high!
Temperature: 26.62  ALARM! Temperature is too high!
Temperature: 27.00  ALARM! Temperature is too high!
Temperature: 27.31  ALARM! Temperature is too high!
```

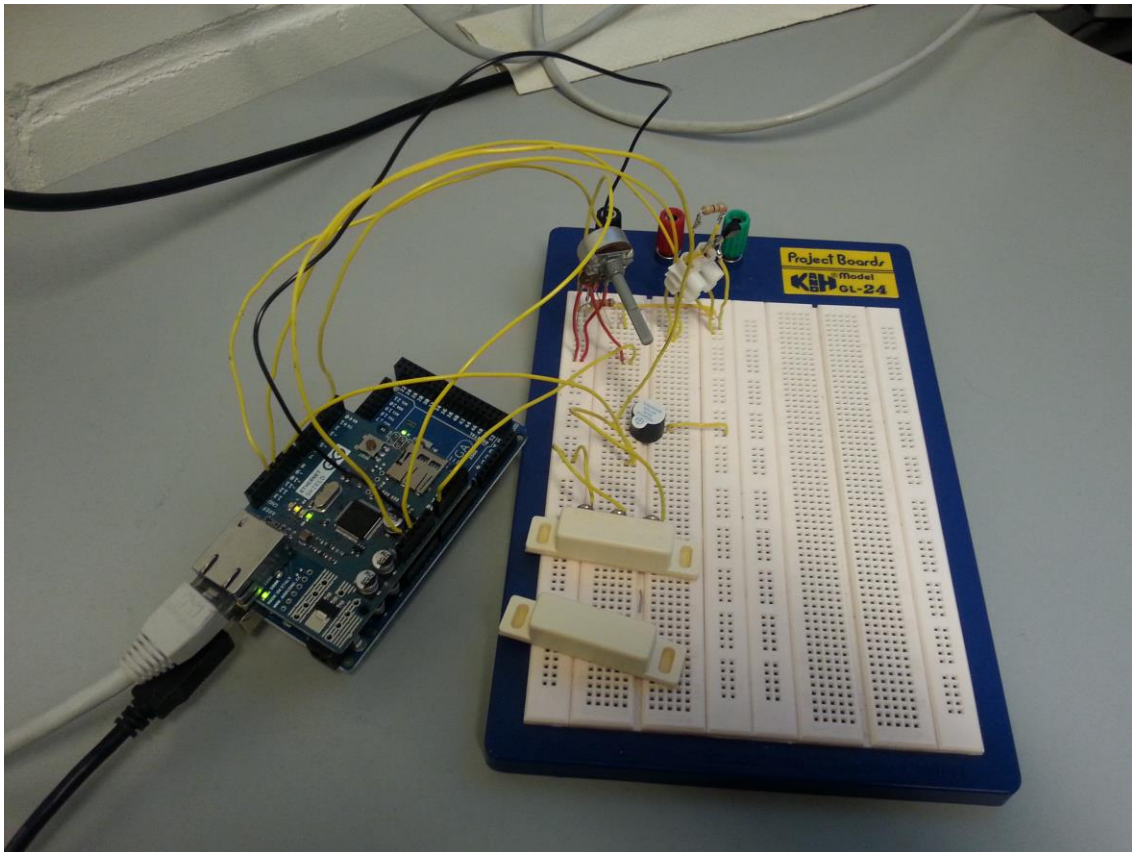
KUVA 6. Lämpötilan tulostaminen PuTTYssä

Kun PuTTY-yhteys oli saatu toimimaan, pääsin tekemään omaa opinnäytetyön aihetta, jossa tarkoituksena oli saada aikaiseksi jääkaapin toimintoja seuraava ja kontrolloiva laite. Ensimmäisenä tehtävänä oli valmistaa hälytysjärjestelmä. Kun lämpötila nousee liian korkeaksi, käynnistyy hälytys, jolloin nettiselaimelle tulostuu hälytysviesti (kuva 7). Samanaikaisesti käynnistyy sekä valo- että äänihälytys piirilevyllä. Valohälytykseen käytin punaista LED-valoa, joka välkkyi sekunnin välein. Myös äänihälytys kuuluu sekunnin välein, jossa käytin piezosummeria.



KUVA 7. Hälytysviestejä nettiselaimella

Koska piirilevyiltä alkoi loppua tila kesken, asensin sensorit ylimääräiselle koekytkentäalustalle (kuva 8). Tällöin pystyin jakamaan +5 V ja +3,3 V jännitteet sekä maadoituksen tulot kaikille sensoreille, kun itse piirilevyllä on ainoastaan yksi paikka +5 V ja +3,3 V jännitteille sekä kolme paikkaa maadoitukselle.



KUVA 8. Arduino on yhdistettynä koekytentäalustaan

Tämän jälkeen lisäsin käyttöliittymään eli nettiselaimelle mahdollisuuden LED-valon ja äänisensorin kontrollointiin. Kun nettiselaimelta kohdasta Light control painetaan ON-painiketta, syttyy LED-valo palamaan. Kun taas painetaan OFF-painiketta, valo sammuu. Tällä ei varsinaisesti ole merkitystä lopullisessa tuotteessa, mutta tämä tehtiin havainnollistamaan esimerkiksi jääkaappiin tulevan sähkövirran katkaisemista. Esimerkiksi jääkaappi voitaisiin sammuttaa netin välityksellä, jos se olisi mökillä ja olisi unohtunut päälle eikä sitä ei käytettäisi pitkään aikaan. Näin säästytään sähkökuluilta eikä jääkaappia tarvitse lähteä erikseen sammuttamaan.

5.3 Ohjelman pilkkomista ja ovikytken asentaminen

Seuraavaksi pilkoin ohjelmakoodin pienempiin palasiin eri välilehdille, koska aiemmin koko koodi oli yhdessä välilehdessä. Tämä helpottaa ohjelman ymmärtämistä. Ilman pilkkomista koodia olisi ollut lähes 1 000 riviä, jolloin olisi ollut todella hidasta ja raskasta etsiä jokin tietty asia koodista. Ohjelma jaettiin

siis itse pääohjelmaan, netti-, measurement- ja clock-välilehtiin. Pääohjelmasta löytyvät setup()- ja loop()-toiminnot (Liite 1), netti-välilehdellä luodaan Internet-sivun ja PuTTY-ohjelman sisältö (Liite 2), measurement-välilehdellä määrätään toiminnot sensoreille (Liite 3) ja clock-välilehdellä luodaan kello eri käyttötarpeita varten (Liite 4).

Kun ohjelma oli saatu toimimaan täydellisesti pilkkomisen jälkeen, lisättiin ohjelmaan oven aukaisua mittaava sensori. Sensorissa on kaksi magneettista osaa, jotka liitetään jääkaapin runkoon ja oven reunaan. Kun ovi aukaistaan, sensorin sisällä oleva kytkin aktivoituu, jolloin sähkövirta pääsee kiertämään sensorin sisällä. Tällöin ohjelma havaitsee, että ovi on auki ja tulostaa Internet-sivulle, että ovi on auki. Lisäsin tähän myös ajastuksen, eli kun ovi on ollut liian kauan auki, tulostuu Internet-sivulle varoitus. Varoituksessa kerrotaan, että ovi on ollut kyseisen ajan verran auki.

5.4 Tietokanta

Ohjelmaan oli tarkoitus lisätä tietokanta, johon tallennettaisiin lämpötilan, säätövastuksen, sähköjännitteen, -virran ja -tehon arvot. Tietokannasta olisi tulostettu nämä arvot graafisena kuviona Internet-sivulle, josta olisi voinut seurata jääkaapin toimintaa.

Tämä ei kuitenkaan onnistunut täydellisesti, koska ohjelma ei osaa lähettää automaattisesti arvoja tietokantaan. Tavoitteena oli siis lähettää arvot sekunnin väliajoin tietokantaan, mutta jostain syystä tämä ei toiminut. Ohjelma osaa kuitenkin lähettää tiedon tietokantaan, kun arvo syötettiin käsin.

Tietokantaohjelmana otettiin käyttöön Wampserver, joka on suunniteltu Windows-pohjaisten tietokoneiden käyttöön. Wampserver on kätevä työkalu, johon on helppo lisätä tietokantoja (11.)

Työssämme lisättiin measurement-niminen taulukko Wampserverille, johon lisättiin aiemmin mainitut kohdat. Seuraavaksi lisättiin PHP-koodilla tehdyt tiedostot Wampserverin www-nimiseen kansioon, josta tiedot tulevat tietokantaan. Viimeisenä lisättiin Arduino-koodiin komento, jolla lähetetään

tiedot PHP-tiedostoon ja siitä tietokantaan. Alla ovat kuvat Arduinossa tehtävästä komennosta (kuva 9) sekä PHP-koodista (kuva 10).

```
153 //      client1.print("<form action='http://localhost/php/new.php' method='post'>");
154 //      client1.print("<input name='voltage' type='text'>");
155 //      client1.print("229.50");
156 //      client1.print("</input>");
157 //      client1.print("</form>");
```

KUVA 9. Arduinossa tehtävä komento

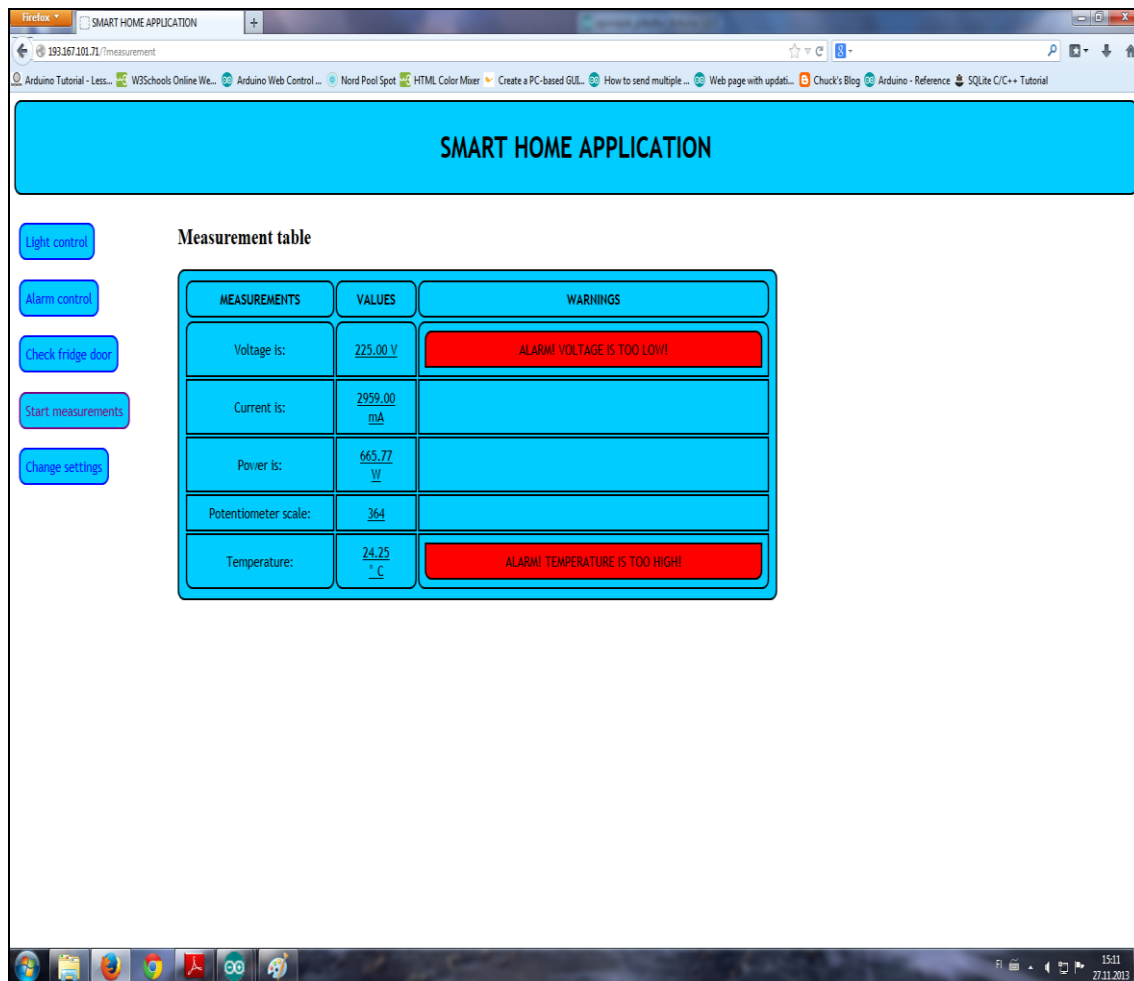
```
1  <?php
2  $link = mysql_connect('localhost', 'php', 'data');
3  if (!$link)
4  {
5      echo $link;
6      echo 'Connection failed';
7      exit();
8  }
9  else if (!mysql_select_db('php', $link))
10 {
11     echo 'Connect to database failed';
12     exit();
13 }
14 else
15 {
16     $voltage = $_GET['voltage'];
17     $current = 11;
18     $power = 12;
19     $potentiometer = 13;
20     $temperature = 14;
21     /*$current = $_GET["current"];
22     $power = $_GET["cPower"];
23     $potentiometer = $_GET["potentiometer"];
24     $temperature = $_GET["temperature"];*/
25
26     echo "your voltage is ". $voltage ." V \n";
27     echo "your current is ". $current ." mA \n";
28     echo "your power is ". $power ." W \n";
29     echo "your potentiometer is ". $potentiometer ."\n";
30     echo "your temperature is ". $temperature ."\n";
31
32     $Sql = "INSERT INTO measurement(Voltage, Current, Power, Potentiometer, Temperature)
33     VALUES ('".$voltage."', '".$current."', '".$power."', '".$potentiometer."', '".$temperature."')";
34     mysql_query($Sql);
35     if (!mysql_query($Sql))
36     {
37         die('Error: ' . mysql_error($link));
38     }
39     echo "1 record added";
40 }
41 -?>
```

KUVA 10. PHP-tiedosto

5.5 Lopullinen tuote

Viimeiseksi muutin Internet-sivun ulkoasua, jotta se näyttäisi hieman hienommalta. Tätä ei varsinaisesti olisi minun tarvinnut tehdä, mutta tarvitsin joitakin valikoita selkeyttämään ohjelman käyttämistä. Tämän jälkeen en muuttanut ulkoasua enää, joten tämä on lopullinen versio Internet-sivusta (kuva 11).

Lopullisessa versiossa on erikseen valo- ja äänihälytysten päälle ja pois kytkeminen. Tämä ei kuitenkaan sammuta hälytyksiä mittauksista eikä jääkaapin oven seurannasta, vaan tämä rakennettiin kuvaamaan jääkaapin virran kytkemistä päälle ja pois. Mittauksiin kuuluvat sähköjännite, -virta ja -teho, sekä säätövastuksen ja lämpötilan arvot. Lisäksi sivulta voi seurata jääkaapin oven tilaa, onko se auki ja kuinka kauan se on ollut auki, jolloin sivulle tulostuu varoitus. Lisäksi sivulle lisättiin asetusvalikko, josta olisi pitänyt pystyä vaihtamaan hälytysajan alkamiskohtaa, mutta se ei onnistunut, koska siihen olisi tarvittu Javascriptiä, jota Arduinon ohjelmointiympäristö ei tukenut. Asetusvalikossa ilmeni toinenkin ongelma, jonka vuoksi sinne ei voinut lisätä tekstikenttiä enempää. Luulen sen johtuvan Ethernet Shieldin muistikapasiteetista.



KUVA 11. Lopullinen versio Internet-sivusta

6 TULOKSET JA POHDINTA

Työ oli haastava, mutta samalla mielenkiintoinen, koska uskon, että tämä aihealue tulee yleistymään lähitulevaisuudessa todella paljon. Uskomukseni perustuu siihen näkökulmaan, että nykyään kaikkialla puhutaan sekä taloudellisuudesta että sähkönkulutuksesta. Tämä olisi ratkaisu näihin molempiin asioihin, koska älykkäällä sähköverkolla saavutettaisiin jopa 30 prosenttia alhaisempi energiankulutus (12). Samalla se jakaisi sähkönkulutuksen tasaisemmin ympäri vuorokautta. Tällöin ei tulisi sähkönkulutuspiikkiä tai ainakaan niin suurta, jolloin sähköyhtiöt joutuvat käyttämään ylimääräisiä voimavaroja sähköntuotannossa. Ohjelmasta olisi apua henkilöasiakkaille myös siinä, että tähän olisi mahdollista rakentaa laitteiden ennakoiva huoltotarpeen ilmoitus. Tällöin laitteet eivät kerkeäisi rikkoutua lopullisesti, vaan ne saataisiin kuntoon huoltotoimenpiteillä. Huonossa kunnossa olevat laitteet vievät myös enemmän sähköä, joka tarkoittaa lisäkuluja sähkölaskuissa.

Työni keskittyi ainoastaan jääkaapin seuraamiseen, joten tästä voisi kehitellä ideaa eteenpäin liittämällä ohjelmaan myös muita laitteita. Kun ohjelmaan lisättäisiin vielä mahdollisuus käyttää sitä julkisen Internetin välityksellä, mahdollistaisi se laitteiden sammuttamisen etäältä, jolloin laitteet eivät vie sähköä, kun ohjelman käyttäjä olisi esimerkiksi lomamatkalla. Tästä voisi kehitellä myös matkapuhelinohjelman, joka antaisi ilmoituksen matkapuhelimeen, kun johonkin laitteeseen tulisi jokin vika tai ongelmatekijä havaittaisiin.

Haastavan työstä teki se, että itselläni ei ollut minkäänlaista kokemusta Arduinosta ja lähinnä Arduino-ohjelmointikieleen kohdistui suurin osa ongelmista, joita riitti työssäni. Ongelmista johtuen en saanut kaikkea valmiiksi, mitä olimme suunnitelleet toteutettavaksi. Tämän vuoksi myös suunniteltu aikataulu muuttui kuukautta pidemmäksi.

Ensimmäisenä työssäni opettelin, miten Arduinon ohjelmointikieli käyttäytyy ja miten sitä kirjoitetaan. Arduinon ohjelmointikieli perustuu kylläkin C++-

kieleen, mutta siinä oli paljon rajoittavia tekijöitä. Suurimmaksi haasteeksi osoittautui muiden ohjelmointikielien käyttö tässä ohjelmointiympäristössä. Joidenkin ohjelmointikielien, kuten HTML-kielen, lisääminen onnistui suurimmaksi osaksi, mutta sekin piti niin sanotusti soluttaa Arduinon oman ohjelmointikielen sisälle. Myös Arduinon muistikapasiteetti tuli vastaan, sillä Internet-sivulle ei pystytty tulostamaan kaikkea haluttua. Arduino soveltuu todella hyvin yksinkertaisten projektien luontiin ja niihin löytyy hyvin myös apua, mutta kun projekti kasvaa suuremmaksi ja vaativammaksi, tulevat rajat vastaan.

Jatkossa olisi hyvä kehittää oma piirilevy, johon suunniteltaisiin kaikki tarvittavat komponentit, liitännät ja tarpeeksi muistikapasiteettia. Siihen voisi lisätä myös mahdollisuuden ylimääräisten komponenttien asentamiseen. Jatkokehittämistä ajatellen olisi hyvä tehdä ohjelmointi tavallisella C-kielellä tai Javalla, jolloin muiden kielten lisääminen onnistuisi paljon helpommin ja koodista tulisi näin selkeämpää. Itse suosisin Javan käyttöä, koska Internet-sivun tekeminen olisi paljon helpompaa sillä ja myös tietokannan tekemiseen löytyy paljon enemmän esimerkkejä Javalla.

Työhön ei pystytty lisäämään myöskään jääkaapin sähkönkulutusta mittaavaa laitetta, koska sen toteuttaminen oli toisella opiskelijalla vielä kesken. Olisi ollut hienoa päästä kokeilemaan, miltä lukemat todellisuudessa näyttäisivät. Myös jääkaapin toimintoja olisi ollut mukava päästä testaamaan, eli toimisiko rakentamani virrankatkaisu todellisuudessa.

LÄHTEET

1. HomePlug Powerline Alliance 2013. Smart Energy Initiative.
http://www.homeplug.org/tech/smart_energy/. Hakupäivä 14.12.2013.
2. Energiateollisuus 2013. Älykäs verkko eli Smart Grid.
<http://energia.fi/sahkomarkkinat/sahkoverkko/alykas-verkko>. Hakupäivä 15.12.2013.
3. Sähkön hinta tunneittain.
 - a. Pekka Pantsu 2012. Halpaa sähkönhintaa tavoitteleva saunoo tunti hinnoittelun mukaan
http://yle.fi/uutiset/halpaa_sahkonhintaa_tavoitteleva_saunoo_tuntihinnoittelun_mukaan/6332134. Hakupäivä 16.9.2013.
 - b. EET 2013. Kuluttaja hyöttyy sähkön tunti hinnoittelusta.
<http://www.fortum.com/fi/media/ajankohtaista/artikkelit/pages/kuluttaja-hyotyy-sahkon-tuntihinnoittelusta.aspx>. Hakupäivä 16.9.2013.
 - c. Nordpoolspot 2013. Elspot price for area Finland
<http://www.nordpoolspot.com/Market-data1/Elspot/Area-Prices/ALL1/Hourly/>. Hakupäivä 11.12.2013.
4. Energiateollisuus 2013. Sähkökatkot ja jakelun keskeytykset.
<http://energia.fi/sahkomarkkinat/sahkoverkko/sahkokatkot-ja-jakelun-keskeytykset>. Hakupäivä 15.12.2013.
5. Energiateollisuus 2013. Hajautettu pientuotanto.
<http://energia.fi/sahkomarkkinat/sahkoverkko/pientuotanto>. Hakupäivä 15.12.2013.
6. Tietoturva.
<http://www.internetopas.com/yleistietoa/tietoturva/>. Hakupäivä 11.12.2013
7. Arduinon toiminta.
<http://arduino.cc/en/>. Hakupäivä 22.11.2013.
8. Arduinon historia.
<http://en.wikipedia.org/wiki/Arduino>. Hakupäivä 22.11.2013.
9. Setup()-toiminto.
<http://arduino.cc/en/Reference/Setup>. Hakupäivä 9.12.2013.

10. Loop()-toiminto.

<http://arduino.cc/en/Reference/Loop>. Hakupäivä 9.12.2013.

11. Tietokantaohjelma.

<http://www.wampserver.com/en/>. Hakupäivä 30.9.2013.

12. ABB. Smart Grid

[http://www02.abb.com/global/seitp/seitp202.nsf/0/6c7d76b1276269fac1257877002a465e/\\$file/ABB_smartgrid_final.pdf](http://www02.abb.com/global/seitp/seitp202.nsf/0/6c7d76b1276269fac1257877002a465e/$file/ABB_smartgrid_final.pdf). Hakupäivä 17.12.2013.

LIITTEET

Liite 1 Ohjelmakoodin pääsivu

Liite 2 Netti.c

Liite 3 Measurement.c

Liite 4 Clock.c

Liite 5 Liitteen otsikko

Liite 6 Liitteen otsikko

```
#include <Timer.h>
```

```
#include <SPI.h>
```

```
#include <Ethernet.h>
```

```
#include <OneWire.h>
```

```
#include <DallasTemperature.h>
```

```
#include <Time.h>
```

```
#include "netti.h"
```

```
// Data wire is plugged into pin 2 on the Arduino
```

```
#define ONE_WIRE_BUS 2
```

```
boolean incoming = 0;
```

```
char osoite;
```

```
void setup()
```

```
{
```

```
    pinMode(redPin, OUTPUT); // sets the digital pin as output
```

```
    pinMode(soundPin, OUTPUT); // sets the digital pin as output
```

```
    pinMode(magnetPin, OUTPUT); // sets the digital pin as output
```

```
    Ethernet.begin(mac, ip, gateway, subnet); // initialize the Ethernet device
```

```
Serial.begin(9600); // enable serial data print

t.every(1000, Time); // every 1 second check if any new data

sensors.begin(); // Start up the library

}
```

```
void loop()

{

  while(1)

  {

    Menu();

    t.update();

    Serial.flush();

    while(!Serial.available()); //stop program until get byte

    {

      t.update();

      char incByte = Serial.read();

      switch(incByte)

      {

        case 'a':

          while('a')

          {
```

```
Serial.println("valitse 1(Voltage), 2(Current), 3(Power) tai r(return)");
```

```
Serial.println("");
```

```
while(!Serial.available());
```

```
{
```

```
    t.update();
```

```
    if (Serial.available() > 0) //check serial if any data
```

```
    {
```

```
        char inByte = Serial.read();
```

```
        t.update();
```

```
        switch(inByte)
```

```
        {
```

```
            case '1': //button 1 gives voltage
```

```
                Serial.print("Voltage: ");
```

```
                Serial.print(Voltage);
```

```
                Serial.println(" V");
```

```
                Serial.flush();
```

```
                break;
```

```
            case '2': //button 2 gives current
```

```
                Serial.print("Current: ");
```

```
                Serial.print(Current);
```

```
Serial.println(" mA");
```

```
Serial.flush();
```

```
break;
```

```
case '3':
```

```
Volt = Voltage / 1000;
```

```
Power = Volt * Current;
```

```
Serial.print("Power: ");
```

```
Serial.print(Power);
```

```
Serial.println(" W");
```

```
Serial.flush();
```

```
break;
```

```
/*case '4': // No need but could make it for later version
```

```
Elenergy = Power * aika;
```

```
Serial.print("Electrical energy: ");
```

```
Serial.print(Elenergy);
```

```
Serial.println(" kW/h");
```

```
Serial.flush();
```

```
break;*/
```



```
case 'r':
```

```
    Serial.flush();
```

```
    return;
```

```
    break;
```

```
default:
```

```
    Serial.flush();
```

```
}
```

```
}
```

```
}
```

```
}
```

```
break;
```

```
case('b'):
```

```
    IPcheck();
```

```
    break;
```

```
case('c'):
```

```
    Potentiometer();
```

```
    break;
```

```
/*case('c'):
```

```
    SetupServer();
```

```
    break;*/
```

```
case('d'):
```

```
    Temperature();
```

```
    break;
```

```
case('e'):
```

```
    while('e')
```

```
    {
```

```
        Serial.println("Valitse 1(LED on), 2(LED off) or r(return)");
```

```
        while(!Serial.available());
```

```
        {
```

```
            t.update();
```

```
            if (Serial.available() > 0) //check serial if any data
```

```
            {
```

```
                char inByte = Serial.read();
```

```
                t.update();
```

```
                switch(inByte)
```

```
{  
  
  case '1': //button 1 gives voltage  
  
    digitalWrite(redPin, HIGH);  // sets the Red LED on  
  
    Serial.println("LED on");  
  
    Serial.println("");  
  
    Serial.flush();  
  
    break;  
  
  case '2':  
  
    digitalWrite(redPin, LOW);  // sets the Red LED off  
  
    Serial.println("LED off");  
  
    Serial.println("");  
  
    Serial.flush();  
  
    break;  
  
  case 'r':  
  
    Serial.flush();  
  
    return;  
  
    Serial.println("");  
  
    Serial.println("");  
  
    break;  
  
  default:
```

```
        Serial.flush();

    }

}

}

}

break;


case('f'):

    while('f')

    {

        EthernetClient client1 = server1.available(); //listen for incoming clients

        if(client1.available())

        {

            Netti();

        }

        if(Serial.read() == 'r')

        {

            break;

        }

    }

}
```

```
    }

    }

}

}

void Menu()

{

    Serial.println("");

    Serial.println("Testi menu");

    Serial.println("Choose from:");

    Serial.println("a: Measurement");

    Serial.println("b: IP check");

    Serial.println("c: Potentiometer");

    Serial.println("d: Temperature");

    //Serial.println("c: Setup server");

    Serial.println("e: LED");

    Serial.println("f: Ethernet");

    Serial.println("");

    Serial.println( "Info: First do IP check then start Ethernet. After starting
Ethernet use the IP Address for PuTTY or web browser. ");

    Serial.println("Info: For PuTTY you must choose Telnet with port 23");

    Serial.println("Info: Go back with letter 'r' ");
```

```
Serial.println("");  
  
}  
  
void IPcheck() //IP-address check  
  
{  
  
  EthernetClient client;  
  
  if (Ethernet.begin(mac) == 0) // start the Ethernet connection:  
  
  {  
  
    Serial.println("Failed to configure Ethernet using DHCP");  
  
    for(;;); // no point in carrying on, so do nothing forevermore:  
  
  }  
  
  Serial.print("My IP address: ");  
  
  Serial.print(Ethernet.localIP());  
  
  if (!client.connected())  
  
  {  
  
    while(1)  
  
    {  
  
      client.stop();  
  
      Serial.println("");  
  
      return;  
  
      //for(;;);  
  
    }  
  
  }  
  
}
```

```
    /*if(Serial.read() == 'r')

    {

        Serial.println("");

        break;

    }*/

}

/*Serial.print("This IP address: "); //optional ip check, need to add byte server
so this would work

IPAddress myIPAddress = Ethernet.localIP();

Serial.print(myIPAddress);

if(client.connect(server, 80)>0) {

    Serial.println(" connected");

    client.println("GET /search?q=arduino HTTP/1.0");

    client.println();

} else {

    Serial.println("connection failed");

}*/

}

// Idea was to make manually server to specific IP address but it seemed this
wasn't needed because void Netti() makes the server by itself
```

```
/*void SetupServer() //Making server to address x

{

Ethernet.begin(mac, osoite); //initialize the ethernet device

server.begin(); //start listening for clients

IPAddress ip(osoite); //defines ip address


Serial.print("Server is at ");

Serial.println(Ethernet.localIP());

Serial.println("");


//EthernetServer server = EthernetServer(80); // create server to port x

}*/
```



```
#include <OneWire.h>
```

```
#include <DallasTemperature.h>
```

```
#include <Timer.h>
```

```
#include <SPI.h>
```

```
#include <Ethernet.h>
```

```
#include <DB.h>
```

```
#include "measurement.h"
```

```
// Data wire is plugged into pin 2 on the Arduino
```

```
#define ONE_WIRE_BUS 2
```

```
extern unsigned int s, m, h;
```

```
long Power;
```

```
float cPower; // Client1 Power
```

```
float Volt;
```

```
float temperature;
```

```
int i;
```

```
boolean gotAMessage = false; // whether or not you got a  
message from the client yet
```

```
String readString; // string for fetching data from address
```

```
EthernetServer server1 = EthernetServer(80); // create server to port x
```

```
EthernetServer server2 = EthernetServer(23);    // create server to port x

byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x46, 0x1E }; // physical mac address

byte ip[] = { 193, 167, 101, 75 };              // ip in lan

byte server[] = { 193,167,101,129 };            // server's address

byte gateway[] = { 193, 167, 101, 254 };        // Internet access via router

byte subnet[] = { 255,255,255,0 };              // subnet mask address for shield

int hours;

int minutes;

Timer seconds;

int changeAlarm = 10;

int changeAlarmTime;


void Netti() //Ethernet server

{

    int sensorValue;

    EthernetClient client1 = server1.available(); //listen for incoming clients

    EthernetClient client2 = server2.available(); //listen for incoming clients

    if (client1)

    {

        Serial.println("new client");
```

```
    boolean currentLineIsBlank = true;           // an http request ends with a
blank line
```

```
    while(client1.available())

    {

        if(client1.available())

        {

            char c = client1.read();

            if(readString.length() < 30)

            {

                readString.concat(c);

            }

            Serial.write(c);

            if (c == '\n' && currentLineIsBlank)

            {

                Serial.println(readString);

                // send a standard http response header

                client1.println("HTTP/1.1 200 OK");    //send new page

                client1.println("Content-Type: text/html image/jpeg");

                client1.println("Connection: close");  // the connection will be closed
after completion of the response

                //      client1.println("Refresh: 1"); // refresh the page automatically every 5
sec
```

```
client1.println();

client1.println("<!DOCTYPE HTML>");

client1.println("<html>");

client1.println("<head>");

client1.println("<title>");

    client1.println("SMART HOME APPLICATION");

client1.println("</title>");

//    client1.println("<script type='text/javascript>");

//        client1.println(" #textInput{ changeAlarm = changeAlarmTime; } ");

//    client1.println("</script>");

client1.println("<style type='text/css'>");


//////// Styling web page //////////

client1.println(" #textInput{ changeAlarm = changeAlarmTime; } ");

client1.println("#tab { width:100%; height:100%; }");

client1.println("#td1 { width:100px; height:100%; border-bottom:2px
solid; }");

    client1.println("#header_bar { width:900px; height:100%; border-
bottom:2px solid; text-align:left; margin:0px 500px 0px 0px; padding:0px 100px
0px 0px; }");

    client1.println("#menu_bar { width:100px; height:100%; padding:40px
20px 10px 10px; text-align:left; }");

    client1.println("#menu { position:absolute; top:150px; left:15px; }");
```

```
client1.println("#text_area { width:900px; height:100%; }");
```

```
client1.println("#text { position:absolute; top:120px; left:250px; }");
```

```
// a = buttons
```

```
client1.println("#a { text-decoration:none; width:75px; height:50px;  
border-color:black; border-top:2px solid; border-bottom:2px solid; border-  
right:2px solid; border-left:2px solid; border-radius:10px 10px 10px; -o-border-  
radius:10px 10px 10px; -webkit-border-radius:10px 10px 10px; font-  
family:'Trebuchet MS',Arial, Helvetica, sans-serif; -moz-border-radius:10px 10px  
10px; background-color:#00CCFF; padding:8px; text-align:center; } ");
```

```
// t = Table fields
```

```
client1.println("#t { border-color:black; border-top:2px solid; border-  
bottom:2px solid; border-right:2px solid; border-left:2px solid; border-  
radius:10px 10px 10px; -o-border-radius:10px 10px 10px; -webkit-border-  
radius:10px 10px 10px; font-family:'Trebuchet MS',Arial, Helvetica, sans-serif; -  
moz-border-radius:10px 10px 10px; background-color:#00CCFF; padding:8px;  
text-align:center; } ");
```

```
client1.println("#t1 { border-color:black; border-top:2px solid; border-  
bottom:2px solid; border-right:2px solid; border-left:2px solid; border-top-left-  
radius:10px; border-top-right-radius:10px; -o-border-top-right-radius:10px; -o-  
border-top-left-radius:10px; -webkit-border-top-left-radius:10px; -webkit-border-  
top-left-radius:10px; font-family:'Trebuchet MS',Arial, Helvetica, sans-serif; -  
moz-border-radius:10px 10px 10px; background-color:#00CCFF; padding:8px;  
text-align:center; } ");
```

```
client1.println("#t2 { border-color:black; border-top:2px solid; border-  
bottom:2px solid; border-right:2px solid; border-left:2px solid; background-  
color:#00CCFF; padding:8px; text-align:center; } ");
```

```
client1.println("#t3 { border-color:black; border-top:2px solid; border-  
bottom:2px solid; border-right:2px solid; border-left:2px solid; border-bottom-left-  
radius:10px; border-bottom-right-radius:10px; -o-border-bottom-right-  
radius:10px; -o-border-bottom-left-radius:10px; -webkit-border-bottom-left-  
radius:10px; -webkit-border-bottom-right-radius:10px; font-family:'Trebuchet  
MS',Arial, Helvetica, sans-serif; -moz-border-radius:10px 10px 10px;  
background-color:#00CCFF; padding:8px; text-align:center; } ");
```

```
// w = Warning fields
```

```
client1.println("#w1 { border-color:black; border-top:2px solid; border-  
bottom:2px solid; border-right:2px solid; border-left:2px solid; border-top-left-  
radius:10px; border-top-right-radius:10px; -o-border-top-right-radius:10px; -o-  
border-top-left-radius:10px; -webkit-border-top-left-radius:10px; -webkit-border-  
top-left-radius:10px; font-family:'Trebuchet MS',Arial, Helvetica, sans-serif; -  
moz-border-radius:10px 10px 10px; background-color:#FF0000; padding:8px;  
text-align:center; } ");
```

```
client1.println("#w2 { border-color:black; border-top:2px solid; border-  
bottom:2px solid; border-right:2px solid; border-left:2px solid; background-  
color:#FF0000; padding:8px; text-align:center; } ");
```

```
client1.println("#w3 { border-color:black; border-top:2px solid; border-  
bottom:2px solid; border-right:2px solid; border-left:2px solid; border-bottom-left-  
radius:10px; border-bottom-right-radius:10px; -o-border-bottom-right-  
radius:10px; -o-border-bottom-left-radius:10px; -webkit-border-bottom-left-  
radius:10px; -webkit-border-bottom-right-radius:10px; font-family:'Trebuchet  
MS',Arial, Helvetica, sans-serif; -moz-border-radius:10px 10px 10px;  
background-color:#FF0000; padding:8px; text-align:center; } ");
```

```
client1.println("#changeAlarm { text-align:right; } ");
```

```
client1.println("</style>");
```

```
client1.println("</head>");

client1.println("<body>");

client1.print("<table id='tab'>");

    client1.print("<div id='t'>");

        client1.println("<h1>SMART HOME APPLICATION</h1>");

    client1.print("</div>");

    client1.print("<td id='menu_bar'>");

        client1.print("<div id='menu'>");

//            client1.println("<a id='a' href='\"/?home\"'>Home page</a><br
/><br /><br />");

            client1.println("<a id='a' href='\"/?light\"'>Light control</a><br
/><br /><br />");

            client1.println("<a id='a' href='\"/?alarm\"'>Alarm control</a><br
/><br /><br />");

            client1.println("<a id='a' href='\"/?door\"'>Check fridge
door</a><br /><br /><br />");

            client1.println("<a id='a' href='\"/?measurement\"'>Start
measurements</a><br /><br /><br />");

            client1.println("<a id='a' href='\"/?settings\"'>Change
settings</a><br /><br /><br />");

        client1.print("</div>");

    client1.print("</td>");

    client1.print("<td id='text_area'>");
```

```
client1.print("<div id='text'>");

//      if(readString.indexOf("?home") >0)  // checks for on
//
//      {
//
//      client1.println("<h2>Introduction</h2>");
//
//      client1.println("Some text to here...");
//
//      client1.println("");
//
//      client1.println("");
//
//      client1.println("");
//
//      }

if(readString.indexOf("?measurement") >0)  // checks for on
{

    client1.println("<h2>Measurement table</h2>");

    client1.println("<meta http-equiv='refresh' content='1'>"); //
Refresh web page when "Start measurements" is selected

    for (int measurement = 0; measurement < 1; measurement++)

    {

        client1.print("<table id='t'>");

        client1.print("<tr>");

        client1.print("<th id='t' width='200px'>");

        client1.print("MEASUREMENTS");

        client1.print("</th>");

        client1.print("<th id='t' width='100px'>");
```



```
        client1.print("VALUES");

        client1.print("</th>");

        client1.print("<th id='t' width='500px'>");

        client1.print("WARNINGS");

        client1.print("</th>");

        client1.print("</tr>");

        client1.print("<tr>");

        client1.print("<td id='t1'>");

        client1.print("Voltage is: ");

        client1.print("</td>");

        client1.print("<td id='t1'>");

        client1.print("<u>");

        /***** This works but it is not version what we need to
have *****/

//          client1.print("<form
action='http://localhost/php/new.php' method='post'>");

//          client1.print("<input name='voltage' type='text'>");

//          client1.print("229.50");

//          client1.print("</input>");

//          client1.print("</form>");
```

```
/******  
*****/  
  
        client1.print(Voltage);  
  
//        client1.print("<?php  
$_POST['http://localhost/php/new.php?Voltage=");  
  
//        client1.print(Voltage);  
  
//        client1.print("]?>");  
  
        client1.print(" V </u>");  
  
        client1.print("</td>");  
  
        client1.print("<td id='t1'>");  
  
        if(Voltage < 227.50)  
        {  
  
            client1.print("<div id='w1'>");  
  
            client1.print(" ALARM! VOLTAGE IS TOO LOW!");  
  
            //LED();  
  
            client1.print("</div>");  
  
        }  
  
        client1.print("</td>");  
  
        client1.print("</tr>");  
  
  
        client1.print("<tr>");  
  
        client1.print("<td id='t2'>");
```

```
        client1.print("Current is: ");

        //long Cur1 = Current;

        client1.print("</td>");

        client1.print("<td id='t2'>");

        client1.print("<u>");

        client1.print("<div name='Current'>");

        client1.println(Current);

//        client1.println("<?php echo $_POST['Current']; ?>");

        client1.print("</div>");

        client1.print(" mA </u>");

        client1.print("</td>");

        client1.print("<td id='t2'>");

        if(Current < 1000)

        {

            client1.print("<div id='w2'>");

            client1.print(" ALARM! CURRENT IS TOO LOW!");

            //LED();

            client1.print("</div>");

        }

        client1.print("</td>");

        client1.print("</tr>");
```

```
client1.print("<tr>");

client1.print("<td id='t2'>");

    client1.print("Power is: ");

client1.print("</td>");

client1.print("<td id='t2'>");

    Volt = Voltage / 1000;

    cPower = Volt * Current;

client1.print("<u>");

client1.print("<div name='cPower'>");

    client1.println(cPower);

//        client1.println("<?php echo $_POST['cPower']; ?>");

client1.print("</div>");

client1.print(" W </u>");

client1.print("</td>");

client1.print("<td id='t2'>");

    if(cPower < 270)

    {

        client1.print("<div id='w2'>");

        client1.print(" ALARM! POWER IS TOO LOW!");

        //LED();
```

```
        client1.print("</div>");

    }

    client1.print("</td>");

    client1.print("</tr>");

    client1.print("<tr>");

    client1.print("<td id='t2'>");

        client1.print("Potentiometer scale: ");

    client1.print("</td>");

    client1.print("<td id='t2'>");

        sensorValue = analogRead(A0); // Get potentiometer's
value

        //potentiometer = sensorValue;

    client1.print("<u>");

    client1.print("<div name='potentiometer'>");

        client1.println(sensorValue); // Print potentiometer's
value

        //client1.println(potentiometer); // Print potentiometer's
value

    //
    client1.println("<?php echo $_POST['sensorValue'];
?>");

    client1.print("</div>");

    client1.print("</u>");

    client1.print("</td>");
```

```
client1.print("<td id='t2'>");

    if(sensorValue < 100)

    {

        client1.print("<div id='w2'>");

        client1.print(" ALARM! POTENTIOMETER IS TOO

LOW!");

        //LED();

        client1.print("</div>");

    }

    client1.print("</td>");

client1.print("</tr>");


client1.print("<tr>");

    client1.print("<td id='t3'>");

        client1.print("Temperature: ");

    client1.print("</td>");

    client1.print("<td id='t3'>");

        sensors.requestTemperatures(); // Get temperature from

sensor

        client1.print("<u>");

        temperature = sensors.getTempCByIndex(0);

        client1.print("<div name='temperature'>");
```

```
client1.println(temperature); // Print temperature from
sensor

// client1.println("<?php echo $_POST['temperature']; ?>");

client1.print("</div>");

client1.print("&#xb0"); // Print degree symbol

client1.print(" C </u>");

client1.print("</td>");

client1.print("<td id='t3'>");

if(sensors.getTempCByIndex(0) > 24.00)

{

client1.print("<div id='w3'>");

client1.print(" ALARM! TEMPERATURE IS TOO HIGH!");

LED();

client1.print("</div>");

}

client1.print("</td>");

client1.print("</tr>");

client1.print("</table>");

}

}

////////// control arduino's pins //////////
```

else if(readString.indexOf("?light") >0) // if you select light control
from web page

```
{  
  
    client1.println("<h2>Light Control</h2>");  
  
    client1.println("<a id='a' href='\"/?lightoff\"'>Turn off</a>");  
  
    client1.println("<a id='a' href='\"/?lighton\"'>Turn On</a><br  
/><br /><br />");
```

```
    if(readString.indexOf("?lightoff") >0)
```

```
{  
  
    digitalWrite(redPin, LOW); // set led off  
  
    client1.println("LED OFF");  
  
}
```

else if(readString.indexOf("?lighton") >0) // if you select
something other from web page

```
{  
  
    digitalWrite(redPin, HIGH); // set led on  
  
    client1.println("LED ON");  
  
}  
  
}
```

else if(readString.indexOf("?alarm") >0) // if you select alarm
control from web page

```
{  
  
    client1.println("<h2>Alarm Control</h2>");
```



```
client1.println("<a id='a' href='\"/?alarmoff\"'>Turn off</a>");

client1.println("<a id='a' href='\"/?alarmon\"'>Turn On</a><br
/><br /><br />");

if(readString.indexOf("?alarmoff") >0)

{

    digitalWrite(soundPin, LOW); // set pin 4 low

    client1.println("Alarm OFF");

//        Serial.println("Alarm Off");

}

else if(readString.indexOf("?alarmon") >0) // if you select
something other from web page

{

    digitalWrite(soundPin, HIGH); // set pin 4 high

    client1.println("Alarm ON");

//        Serial.println("Alarm On");

}

}

else if(readString.indexOf("?door") >0) // if you select Check
fridge door from web page

{

    client1.println("<h2>Fridge inspection</h2>");
```

```
client1.println("<meta http-equiv=\"refresh\" content=\"1\">"); //  
Refresh web page when "Check fridge door" is selected
```

```
currState = digitalRead(magnetPin); // check position of door(1  
or 0)
```

```
if(currState == 1) // if door is open  
  
{  
  
    client1.println(s);  
  
    client1.println(changeAlarm);  
  
    if(h == 0 && m == 0 && s <= changeAlarm)  
  
    {  
  
        client1.println("DOOR IS OPEN");  
  
        if(readString.indexOf("?door") < 0) // if you select something  
other from web page
```

```
    {  
  
        return; // return to menu  
  
        break; // break for loops  
  
    }
```

```
}
```

```
else
```

```
{  
  
    client1.print("ALARM! DOOR HAVE BEEN OPEN: ");  
  
    client1.print(h);
```

```
        client1.print(" HOURS ");

        client1.print(m);

        client1.print(" MINUTES ");

        client1.print(s);

        client1.println(" SECONDS");

        client1.println("");

        LED();

        if(readString.indexOf("?door") < 0) // if you select something
other from web page

        {

            return; // return to menu

            break; // break for loops

        }

    }

}

else if(currState == 0)

{

    h = 0;

    m = 0;

    s = 0;

    digitalWrite(soundPin, LOW);

    digitalWrite(redPin, LOW);
```

```
        client1.println("DOOR IS CLOSED");

    }

    if(readString.indexOf("?door") <0) // if you select something
other from web page

    {

        return; // return to menu

        break; // break for loops

    }

}

else if(readString.indexOf("?settings") >0) // if you select alarm
control from web page

{

    client1.print("<div>");

//        client1.println("<h2>Change properties</h2>");

//        client1.println("Here you can change settings of fridge door
alarm start time, etc..");

//        client1.println("<h3>Door alarm settings</h3>");

        client1.println("Change time when alarm start:</br>");

//        client1.println("<form action=\"\" method=\"post\"
onsubmit=\"changeAlarmTimeSet(this.value);\">");

//        changeAlarmTime = client1.println("0 <input type=\"range\"
id=\"textInput\" name=\"changeAlarmTime\" min=\"0\" max=\"100\" > 60 </br>");
```

```
changeAlarmTime = client1.println("<input type=\"text\"  
id=\"changeAlarm\" name=\"changeAlarmTime\" min=\"0\" max=\"100\" > s  
</br>");
```

```
client1.println("<input type=\"submit\" name=\"textInput\"  
id=\"textInput\" value=\"Set time\" onClick=changeAlarmTime >");
```

```
// client1.println("</form>");
```

```
// if(readString.indexOf("?settings") >0)
```

```
// {
```

```
// changeAlarm = changeAlarmTime;
```

```
// client1.println(changeAlarm);
```

```
// Serial.println(changeAlarm);
```

```
// }
```

```
// if(onClick=changeAlarmTime >0)
```

```
// {
```

```
// changeAlarm = changeAlarmTime;
```

```
// client1.println(changeAlarm);
```

```
// Serial.println(changeAlarm);
```

```
// }
```

```
client1.println(changeAlarmTime);
```

```
// client1.println("Alarm time is now ");
```

```
client1.println(changeAlarm);
```

```
//          client1.println(" s");

          client1.print("</div>");

      }

      client1.print("</div>");

      client1.print("</td>");

      client1.print("</table>");


      //clearing string for next read

      readString="";

      client1.println("</body>");

      client1.println("</html>");

  }

  else if (c == '\n')

  {

      currentLineIsBlank = true; // you're starting a new line

  }

  else if (c != '\r')

  {

      currentLineIsBlank = false; // you've gotten a character on the current
line

  }

}
```

```

}

delay(1); // give the web browser time to receive the data

client1.stop(); // close the connection:

Serial.println("client disconnected");

Serial.println("");

}

else if (client2) // when the client sends the first byte, say hello:

{

    Serial.println("Hello!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!");

    if (!gotAMessage)

    {

        Serial.println("We have a new client");

        client2.println("Hello, client!");

        gotAMessage = true;

        client2.flush();

    }

    char thisChar = client2.read(); // read the bytes incoming from the client:

    while(thisChar == 'a' || thisChar == 'b' || thisChar == 'c' || thisChar == 'd' )

    {

        client2.flush();

        switch(thisChar)

```

```
{

case 'a':

    client2.println("valitse 1(Voltage), 2(Current), 3(Power) tai r(return)");

    while(!client2.available());

    {

        char inByte = client2.read();

        switch(inByte)

        {

            case '1': //button 1 gives voltage

                client2.print("Voltage: ");

                client2.print(Voltage);

                client2.println(" V");

                client2.flush();

                break;

            case '2': //button 2 gives current

                client2.print("Current: ");

                client2.print(Current);

                client2.println(" mA");

                client2.flush();

                break;
```



```
case '3':
```

```
    Volt = Voltage / 1000;
```

```
    Power = Volt * Current;
```

```
    client2.print("Power: ");
```

```
    client2.print(Power);
```

```
    client2.println(" W");
```

```
    client2.flush();
```

```
    break;
```

```
case 'r':
```

```
    client2.flush();
```

```
    return;
```

```
    break;
```

```
default:
```

```
    client2.flush();
```

```
    break;
```

```
}
```

```
}
```

```
client2.flush();
```

```
break;
```

```
case('b'):
```

```
for(i = 0; i < 26; i++)
```

```
{
```

```
    client2.print("Potentiometer scale: ");
```

```
    sensorValue = analogRead(A0);
```

```
    client2.println(sensorValue);
```

```
    delay(500);
```

```
    if(i == 25)
```

```
    {
```

```
        client2.flush();
```

```
        return;
```

```
    }
```

```
}
```

```
break;
```

```
case('c'):
```

```
for(i = 1; i < 16; i++)
```

```
{
```

```
    client2.print("Temperature: ");
```

```
sensors.requestTemperatures();

client2.println(sensors.getTempCByIndex(0));

if(sensors.getTempCByIndex(0) > 24.00)

{

    client2.println("ALARM! Temperature is too high!");

    LED();

}

delay(1000);

if(i == 15)

{

    client2.flush();

    return;

}

}

break;


default:

break;

}

client2.flush();

}
```

```
    client2.flush();  
  
}  
  
client2.flush();  
  
client2.println("a: Measurement, b: Potentiometer, c: Temperature");  
  
}
```

```
#include <SPI.h>
```

```
#include <Ethernet.h>
```

```
#include <OneWire.h>
```

```
#include <DallasTemperature.h>
```

```
#include <Timer.h>
```

```
#include <Time.h>
```

```
#include "clock.h"
```

```
// Data wire is plugged into pin 2 on the Arduino
```

```
#define ONE_WIRE_BUS 2
```

```
// Setup a oneWire instance to communicate with any OneWire devices (not just  
Maxim/Dallas temperature ICs)
```

```
OneWire oneWire(ONE_WIRE_BUS);
```

```
// Pass our oneWire reference to Dallas Temperature.
```

```
DallasTemperature sensors(&oneWire);
```

```
void LED()
```

```
{
```

```
  while(1)
```

```
{  
  
    digitalWrite(redPin, HIGH);    // sets the Red LED on  
  
    digitalWrite(soundPin, HIGH); // sets the Green LED on  
  
    delay(100);                    // waits for 1/10 a second  
  
    digitalWrite(redPin, LOW);     // sets the Red LED off  
  
    digitalWrite(soundPin, LOW);   // sets the Green LED off  
  
    delay(900);                    // waits for 1/10 a second  
  
    break;  
  
}  
  
}
```

```
void Potentiometer()
```

```
{  
  
    while(1)  
  
    {  
  
        int sensorValue = analogRead(A0); // read the input on analog pin 0:  
  
        Serial.println(sensorValue); // print out the value you read:  
  
        delay(1);    // delay in between reads for stability  
  
        if(Serial.read() == 'r')  
  
        {  
  
            break;  
  
        }  
  
    }  
  
}
```

```
    }

}

}

void Temperature()

{

    while(1)

    {

        sensors.requestTemperatures(); // call sensors.requestTemperatures() to
issue a global temperature

        Serial.print("Temperature is: ");

        Serial.println(sensors.getTempCByIndex(0)); // Print value of temperature to
Serial Monitor

        if(sensors.getTempCByIndex(0) > 24.00) // Alarm will turn on if temperature
risen too high

        {

            Serial.println("ALARM! Temperature is too high!");

            LED();

        }

        if(Serial.read() == 'r')

        {

            break;
```

}

}

}


```
#include <Time.h>
```

```
#include <SPI.h>
```

```
#include <Ethernet.h>
```

```
#include <OneWire.h>
```

```
#include <DallasTemperature.h>
```

```
#include <Timer.h>
```

```
int currState;
```

```
int prevState = LOW;
```

```
unsigned long aika;
```

```
float Voltage;
```

```
float Current;
```

```
unsigned int s;
```

```
unsigned int m;
```

```
unsigned int h;
```

```
int redPin = 12; // Red LED connected to digital pin 12
```

```
int soundPin = 11; // Speaker connected to digital pin 11
```

```
int magnetPin = 9; // Magnet connected to digital pin 9
```

```
Timer t;
```

```
//void DoorClock()
```

```
//{
```

```
// s++;
```

```
// if(s >= 60)
```

```
// {
```

```
// m++;
```

```
// s = 0;
```

```
// }
```

```
// if(m >= 60)
```

```
// {
```

```
// h++;
```

```
// m = 0;
```

```
// }
```

```
// if(h >= 24)
```

```
// {
```

```
// h = 0;
```

```
// }
```

```
//}
```

```
void Time() // initialize timer1
```

```
{
```

```
noInterrupts();      // disable all interrupts

//set timer1 interrupt at 1Hz

TCCR1A = 0; // set entire TCCR1A register to 0

TCCR1B = 0; // same for TCCR1B

TCNT1 = 0; //initialize counter value to 0

// set compare match register for 1hz increments

OCR1A = 15624; // = (16*10^6) / (1*1024) - 1 (must be <65536)

// turn on CTC mode

TCCR1B |= (1 << WGM12);

// Set CS10 and CS12 bits for 1024 prescaler

TCCR1B |= (1 << CS12) | (1 << CS10);

// enable timer compare interrupt

TIMSK1 |= (1 << OCIE1A);

interrupts();      // enable all interrupts

}
```

```
ISR(TIMER1_COMPA_vect)    // timer compare interrupt service routine
```

```
{
```

```
    s++;
```

```
    if(s >= 60)
```

```
    {
```

```
        m++;
```

```
        s = 0;
```

```
    }
```

```
    if(m >= 60)
```

```
    {
```

```
        h++;
```

```
        m = 0;
```

```
    }
```

```
    if(h >= 24)
```

```
    {
```

```
        h = 0;
```

```
    }
```

```
    switch(aika)
```

```
    {
```

case 0:

aika++;

break;

case 1:

aika++;

break;

case 2:

aika++;

break;

case 3:

aika++;

break;

case 4:

aika++;

break;

case 5:

aika++;

break;

case 6:

aika++;

break;

```
case 7:

    aika++;

    break;

case 8:

    aika++;

    break;

case 9:

    aika++;

    break;

case 10:

    Voltage = random(225,230); // Set random number between 225 and 230 to
Voltage

    Current = random(1,10000); // Set random number between 1 and 10000 to
Current

    aika = 0;

    break;

default:

    aika = 0;

    break;

}

}
```